

**REMARKS/ARGUMENTS**

Favorable reconsideration and allowance of the present application are respectfully requested in view of the following remarks. Claims 27-55 were pending prior to the Office Action.

**A. SUMMARY OF THIS AMENDMENT**

By the current amendment, Applicants basically:

1. Editorially amend the specification.
2. Cancel claims 27, 41, 45 and 50-55 without prejudice or disclaimer.
3. Amend claims 28-40, 42-44 and 46-49.
4. Respectfully traverse all prior art rejections.

**B. FORM 1449 ACKNOWLEDGMENT REQUESTED**

In the Office Action, the Examiner states that the Information Disclosure Statement submitted on August 14, 2006 fails to comply with 37 C.F.R. § 1.98(a)(2) requiring a legible copy of each cited foreign patent document. For the Examiner's convenience, clean copies of the cited documents are provided herein. Applicants request consideration of all cited references and an acknowledgement thereof in Form-1449.<sup>1</sup>

---

<sup>1</sup> Applicants note that two of the cited references are used in the current Office Action. Thus, Applicants presume that information contained in these references have been considered.

**C. OBJECTION TO THE DRAWINGS**

Figs. 1-4 are objected to for not including a proper legend. A drawing change request is submitted herewith to designate Figs. 1-4 as the Examiner suggests. Applicants respectfully request that the objection to the drawings be withdrawn.

**D. CLAIM OBJECTION**

Claims 28-44 are objected to for informalities. Objection is moot with respect to claim 41. The remainder of objected claims are amended as the Examiner suggests. Applicants respectfully request that the objection to claims 28-44 be withdrawn.

**E. § 112, 2ND PARAGRAPH REJECTION**

Claims 27-44 stand rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite. Specifically, independent claim 27 includes an insufficient antecedent basis for a feature. As noted above, claim 27 is canceled rendering the rejection moot with respect to independent claim 27. The rejection is also moot with respect to claim 41. The remainder of the rejected claims are amended to address informalities including antecedent basis issues.

Applicants respectfully request that the rejection of claims 27-44 under 35 U.S.C. § 112, second paragraph, be withdrawn.

#### **F. § 101 REJECTION**

Claims 47-55 stand rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. The rejection is moot with respect to claims 50-55. Claims 47-49 are amended to address this issue.

Applicants respectfully request that the rejection of claims 47-55 under 35 U.S.C. § 101 be withdrawn.

#### **G. PATENTABILITY OF THE CLAIMS**

In the Office Action, the Examiner makes the following rejections:

- claims 27, 28, 32-34 and 36-55 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Eggert; and
- claims 29-31 and 35 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Eggert in view of Nikander et al.

With regard to claims 27, 41, 45 and 50-55, the rejections are rendered moot. Applicants respectfully traverse with respect to the remaining claims.

In the Office Action, the Examiner details the rejection of claim 27 as being anticipated by Eggert and states that same rationale is applied to independent claim 46. Claim 46 is directed toward a method for use by a host identity protocol (HIP) proxy node to at least partially secured communication between a first host which is not HIP enabled (e.g. a legacy IPv4 host) and a second host which is HIP enabled (e.g. an IPv6) node. An example is illustrated in Fig. 5 which shows a situation in which a legacy host 12 (not HIP enabled)

wishes to initiate communication between itself and a HIP host 14. A HIP proxy node 16 performs an example inventive method illustrated in Fig. 6 to at least partially secure the communication between the legacy host 12 and the HIP host 14.

As illustrated in Fig. 6, when the legacy host 12 wishes to initiate communication with the HIP host 14, it sends a domain name system (DNS) query to its usual DNS server to resolve the IP address of the HIP host 14. However, instead of being received directly at the DNS server, the HIP proxy node 16 intercepts the query. In other words, the HIP proxy node 16 receives the query from the legacy host 12.

In response to the query from the legacy hosts, the HIP proxy node 16 retrieves an IP address and a host identity tag (HIT) associated with the HIP node 14. In the example method illustrated in Fig. 6, the IP address is retrieved by the HIP proxy node 16 sending a DNS query to the DNS server 24-1, which communicates with DNS server 24-2 to retrieve the IP address and the HIT associated with the HIP host 14. *Specification, page 15, last paragraph.*

Since the HIP proxy node 16 knows that the legacy host 12 is not HIP enabled, the HIP proxy node 16 generates a substitute IP address to be associated with the HIP node 14, and maintains a mapping between the substitute IP address, the retrieved IP address and the retrieved HIT. To the legacy host 12, the HIP proxy node 16 returns the substitute IP address

associated with the HIP node 14. *Specification, page 16, second and third paragraphs.*

When the legacy host 12 is ready to initiate a connection to the HIP node 14, it sends a session initiation message, in which the message includes the substitute IP address as the destination address. Upon receipt of the session initiation message from the legacy host 12, the HIP proxy node 16 uses the mapping to negotiate a secure HIP connection between itself and the HIP node 14. *Specification, page 16, last two paragraphs.*

Note the steps are performed by the HIP proxy node. That is, the HIP proxy node receives the query from the legacy hosts, retrieves the IP address and HIT associated with the HIP node, generates and returns the IP address and HIT associated with the HIP node, generates and returns the substitute IP address to the legacy host, and negotiates a HIP connection between itself and the HIP node when a session initiation message is received from the legacy host. Claim 46 reflects these features. Claim 46 is directed towards a method for use by the HIP node. Thus, it is implicit that all recited steps are performed by the HIP proxy node. However, claim 46 is amended to make explicit features that were already present.

The Examiner relies upon Eggert to allegedly disclose the cited features. However, it is noted that there is no single node in Eggert that performs all of these steps. Indeed, even the collection of the nodes in Eggert do not perform the recited steps collectively.

The Examiner in particular relies upon Section 4 of Eggert which describes communication between HIP and non-HIP nodes. This is also illustrated in Fig. 4. In the Office Action, the Examiner equates the rendezvous servers (RVS) with the claimed HIP proxy node.

In Section 4.1 of Eggert, which describes a non-HIP initiator to a HIP responder scenario, Eggert states that when non-HIP node starts communication with HIP responder, it performs a DNS look-up and receives from the DNS a host identity HI of the responder and an IP address of RVS in return. The RVS takes no part in receiving the initial DNS query from the non-HIP initiator. This is sufficient to distinguish claim 46 from Eggert.

The following is noted as well. The Examiner relies upon Section 4.2 to allegedly teach the step of sending a query from the first host to resolve the Internet protocol address of the second host. It is noted that Section 4.2 describes a scenario in which the HIP enabled node is the initiator and the non-HIP node is the responder, which is not applicable to the claim.

Further, the steps described in Section 4.2 is in reference to Fig. 5, not Fig. 4. But even here, when the initiator (the HIP node) initiates a transport layer connection to the responder, it performs a domain name look-up on the DNS server, and receives the actual IP address of the non-HIP responder. Again, the RVS is not involved in the initial query.

The Examiner then goes back to Fig. 4 and alleges that Eggert teaches the feature of retrieving IP address and host identity tag associated with the

second host in response to a query from the first host. But here again, it is the DNS server which provides the non-HIP initiator with the host identity of the responder and an IP address of the RVS, not the IP address of the responder. RVS is simply not involved.

The Examiner may be equating the IP address of the RVS returned by the domain name server as being equivalent to the substitute IP address of the responder since it acts as an IP address for the responder in place of the actual IP address of the responder, which is denoted as IP(R) in Eggert. Under this interpretation, the address IP(R) of the responder in Eggert must correspond to the "IP address" in claim 46.

However, with such an interpretation, Eggert does not disclose retrieving the IP address in response to the DNS query as required. As clearly shown, the DNS server in Eggert returns the IP address of RVS, referred to as IP(RVS). The actual IP address IP(R) of the responder is updated in RVS whenever IP(R) changes. This is quite different from claim 46.

In addition, claim 46 states that the mapping is maintained between the substitute IP address, the retrieved IP address and the retrieved host identity tag. But as seen in Eggert, RVS only maintains a mapping of the host identity tag and the IP addresses of the responders.

It is to be noted that the arrangement shown in Figure 4 of Eggert is similar to the type of situation which claim 46 address in that a non-HIP node initiates communication with a HIP-enabled node. One such problem

associated with this type of scenario is summarized on page 10 of specification, and relates to the situation where the HIP host is located behind a Forwarding Agent. In that situation, using the scheme shown in Figure 4 of Eggert, the non-HIP host would receive the IP address of the Forwarding Agent from the DNS server, which is of no use without the HIT (which the non-HIP host cannot understand).

In fact, Eggert acknowledges in section 4.3 the same problem with the arrangement of Figure 4 of Eggert as identified in the present disclosure. ("This causes several challenges for end-to-end communication"). However, Eggert does not provide sufficient description to enable a solution to this problem of communication between legacy hosts and HIP-enabled hosts.

In particular, in section 4.3.3, Eggert addresses a problem related with the fact that the RVS server does not receive the HIT of the HIP node from the non-HIP node. Eggert suggests a possible solution to register at the RVS a unique IP address for each HIP node that it serves. This unique IP address is also, according to Eggert, registered at the DNS server. A non-HIP node using the unique IP address would enable the RVS to map this address to the HIP of the HIP proxy. However, a question remains as to what in fact the non-HIP node retrieves from DNS in this scenario: there is now both an IP address of the RVS (registered at DNS) and the unique IP address of the HIP node. Eggert provides no description of how a DNS query would work in this scenario.



It is clear that Eggert does not disclose the concept of a single HIP proxy node according to the invention as claimed in claim 46. Eggert merely discloses a known DNS and a known RVS which is actually analogous to a Forwarding Agent. But in addition to these nodes, the inventive method of claim 46 introduces a HIP proxy node that is HIP enabled, and it is the steps carried out by the HIP proxy node that are claimed along with the HIP proxy node itself.

For at least the above stated reasons, claim 46 is distinguishable over Eggert. Nikander et al. does not correct the deficiencies of Eggert. Thus, independent claim 46 is also distinguishable over Eggert and/or Nikander et al. For similar reasons, independent claim 47 is distinguishable over Eggert and/or Nikander et al. Claims 28-40, 42-44, 48 and 49 are distinguishable over Eggert and/or Nikander et al. by virtue of their dependencies from independent claims as well as on their own merits.

Applicants respectfully request that the rejection of claims based on Eggert and Nikander et al. be withdrawn.

#### **H. CONCLUSION**

All objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the present application is in condition for allowance. Should there be any outstanding matters that need to be resolved, the Examiner is respectfully requested to contact Hyung Sohn

AMENDMENT  
U.S. Application No. 10/589,238

Atty. Docket No.: 3772-34  
Art Unit No.: 2431


(Reg. No. 44,346), to conduct an interview in an effort to expedite prosecution in connection with the present application.

The Commissioner is authorized to charge the undersigned's deposit account #14-1140 in whatever amount is necessary for entry of these papers and the continued pendency of the captioned application.

Respectfully submitted,

**NIXON & VANDERHYE P.C.**

By: \_\_\_\_\_

  
Hyung M. Sohn  
Reg. No. 44,346

HNS/edg  
901 North Glebe Road, 11th Floor  
Arlington, VA 22203-1808  
Telephone: (703) 816-4000  
Facsimile: (703) 816-4100

Network Working Group  
 Internet-Draft  
 Expires: August 5, 2004

XP-002300905

L. Eggert  
 NEC  
 February 5, 2004

D1

Host Identity Protocol (HIP) Rendezvous Mechanisms  
 draft-eggert-hip-rendezvous

P11.05-02-2004

P. 1-23

23

## Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 5, 2004.

## Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

## Abstract

This document discusses rendezvous mechanisms for the Host Identity Protocol (HIP). Rendezvous mechanisms, such as HIP Rendezvous Servers, improve operation when HIP nodes are multi-homed or mobile. They can also facilitate communication between HIP and non-HIP nodes. Possible rendezvous mechanisms differ in performance, compatibility, and impact on the HIP and Internet architectures.

Eggert

Expires August 5, 2004

[Page 1]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

## Table of Contents

1.	Introduction . . . . .	3
2.	Communication Between HIP Nodes . . . . .	5
3.	Communication Between Mobile or Multi-Homed HIP Nodes . . . . .	7
3.1	Mobility and Multi-Homing with DNS Updates . . . . .	7
3.2	Mobility and Multi-Homing with Rendezvous Servers . . . . .	8
4.	Communication Between HIP and Non-HIP Nodes . . . . .	11
4.1	Non-HIP Initiator to HIP Responder . . . . .	11
4.2	HIP Initiator to Non-HIP Responder . . . . .	12
4.3	Discussion . . . . .	13

4.3.1 Relaying Overhead . . . . .	14
4.3.2 Return Traffic . . . . .	14
4.3.3 Node Identification . . . . .	14
4.3.4 Network Address Translation . . . . .	15
5. Rendezvous Broker . . . . .	16
5.1 Comparison to Rendezvous Servers . . . . .	17
5.2 Mobility . . . . .	18
5.3 Tunneling . . . . .	18
6. Security Considerations . . . . .	20
7. Acknowledgments . . . . .	21
Normative References . . . . .	22
Informative References . . . . .	23
Author's Address . . . . .	24
A. Document Revision History . . . . .	25
Intellectual Property and Copyright Statements . . . . .	26

Eggert

Expires August 5, 2004

[Page 2]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

## 1. Introduction

The current Internet uses two global namespaces: domain names and IP addresses. The Domain Name System (DNS) provides a two-way lookup service between the two [1]. Domain names are symbolic identifiers for sets of IP addresses.

IP addresses have two uses. First, they are topological locators for network attachment points. Second, they act as names for the attached network interfaces. Saltzer [13] discusses these naming concepts in detail.

Routing and other network-layer mechanisms are based on the locator aspects of IP addresses. Transport-layer protocols and mechanisms typically use IP addresses in their role as names for communication endpoints.

This dual use of IP addresses limits the flexibility of the Internet architecture. The need to avoid readdressing in order to maintain existing transport-layer connections complicates advanced functionality, such as mobility, multi-homing, or network composition. Sunshine summarizes the consequences of addressing on advanced network functions [14].

The Host Identity Protocol (HIP) architecture [2] defines a new third namespace. The Host Identity namespace decouples the name and locator roles currently filled by IP addresses. Instead of mapping domain names directly into IP addresses, HIP maps domain names into Host Identities, and Host Identities into IP addresses. Transport-layer mechanisms operate on Host Identities instead of using IP addresses as endpoint names. Network-layer mechanisms continue to use IP addresses as pure locators.

Without HIP, nodes establish transport-layer connections by first looking up the fully-qualified domain name (FQDN) of a peer in the DNS. A successful DNS lookup returns the peer's IP addresses. A node uses one of the returned IP addresses to initiate transport-layer communication with a peer node.

HIP nodes will also look up the domain name of desired peers in the DNS. When a successful lookup includes a peer's Host Identities, HIP nodes perform a HIP Base Exchange before establishing transport-layer connections. The HIP Base Exchange authenticates the end hosts and can bootstrap encryption of the subsequent communication with IPsec [15]. The HIP specification [3] discusses the details of the Base Exchange and the related protocol exchanges.

After the Base Exchange, HIP nodes use Host Identities instead of IP

Eggert

Expires August 5, 2004

[Page 3]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

addresses for transport-layer connections with a peer. The HIP layer in the network stack internally translates Host Identities (HI) into network-layer IP addresses. This additional mapping between Host Identities and IP addresses (HI->IP) is logically separate from the first mapping between fully-qualified domain names and Host Identities (FQDN->HI).

For application and transport-layer compatibility, the FQDN->HI mapping must remain in the DNS. However, the HI->IP mapping is internal to the HIP layer and may be performed in a number of ways. Different lookup mechanism may support communication between two mobile or multi-homed HIP nodes better [4].

Transparent communication between HIP and non-HIP nodes places additional restrictions on the lookup mechanisms. For example, non-HIP nodes expect DNS lookups to return IP addresses, requiring the HI->IP mapping (or a representation thereof) to remain in the DNS. Section 4 discusses communication between HIP and non-HIP nodes and describes different alternatives that support it.

Eggert

Expires August 5, 2004

[Page 4]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

## 2. Communication Between HIP Nodes

In the current Internet, the DNS provides a FQDN->IP mapping. With HIP, it must continue to provide a mapping based on domain names. This allows transport-layer connections to bind to Host Identities instead of IP addresses transparently.

Instead of mapping domain names directly into IP addresses (FQDN->IP), with HIP the DNS maps them to Host Identities (FQDN->HI). In a second step, another lookup that is internal to the HIP-layer translates the Host Identities into IP addresses for network-layer delivery (HI->IP).

Several alternative approaches are possible for maintaining the HI->IP information. The DNS can maintain this mapping along with the FQDN->HI mapping. Alternatively, a database separate from the DNS can manage this information. This section discusses the different approaches and their implications on communication between two HIP nodes. Section 4 will discuss the compatibility aspects of the alternatives described here when HIP and non-HIP nodes communicate.

The HIP architecture and protocol specifications suggest storing Host Identities along with a node's IP addresses in the DNS [2] [3]. The index for both tables will be domain names. Logically, the DNS will thus contain two separate mappings: FQDN->HI and FQDN->IP.

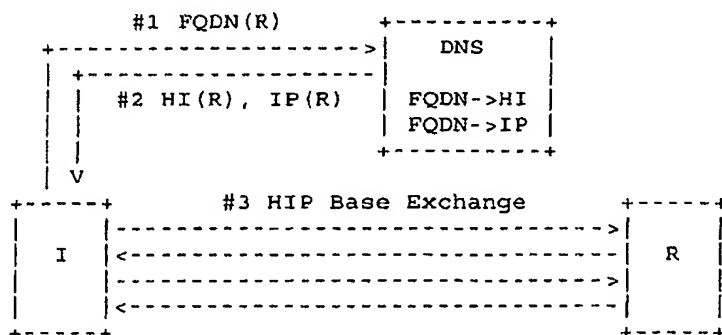


Figure 1: HIP Lookup and Base Exchange

Figure 1 shows the lookup steps and HIP Base Exchange when a node's Host Identities are stored alongside its IP addresses. In step #1, the initiator I performs a DNS lookup on R's domain name FQDN(R). The DNS server responds with both R's Host Identities HI(R) and its IP addresses IP(R) in step #2.

Eggert

Expires August 5, 2004

[Page 5]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

The initiator I uses both pieces of information to perform the HIP Base Exchange with R in step #3. (The details of the Base Exchange, specified in [3], are not relevant to this discussion and will thus be omitted.)

Note that the DNS does not currently store the HI->IP mapping directly. Instead, a DNS lookup on a domain name returns both its FQDN->HI and FQDN->IP entries. The HIP stack then implicitly constructs the HI->IP mapping based on the HI and IP information returned by the DNS lookup. In the example in Figure 1, the FQDN(R) lookup in step #1 returns both HI(R) and IP(R) in step #2. HIP implicitly constructs the HI(R)->IP(R) mapping based on the assumption that HI(R) is reachable at IP(R).

One disadvantage of this approach is that a node's domain name is required to obtain both its Host Identities and its IP addresses. Even if a HIP node already knows the Host Identity of a HIP peer through other means, it cannot currently obtain the peer's IP addresses through the DNS. The DNS does not maintain an explicit HI->IP table, but instead indexes Host Identities only by domain names.

A reverse HIP->FQDN DNS mapping could address this limitation. HIP nodes would then look up a HIP peer's domain name through its Host Identity. They would then use the returned domain name to find the peer's IP addresses in a second lookup. However, the DNS may not be structurally suited to maintain the reverse HIP->FQDN mapping. As the main Internet-wide database, the DNS is already being overloaded with functionality that might be better handled with new mechanisms [16]. Finally, the additional reverse lookup would increase the latency of the HIP Base Exchange.

Eggert

Expires August 5, 2004

[Page 6]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

### 3. Communication Between Mobile or Multi-Homed HIP Nodes

HIP decouples domain names from IP addresses. Because transport

protocols bind to Host Identities, they remain unaware if the set of IP addresses associated with a Host Identity changes. This change can have various reasons, including, but not limited to, mobility and multi-homing.

Proposed extensions for mobility and multi-homing [4] allow a HIP node to notify its peers about changes in its set of IP addresses. These extensions require an established HIP association between two nodes, i.e., a completed HIP Base Exchange.

In addition to notifying its current peers about changes in its IP addresses, a HIP node must also update its HI->IP mapping in response to IP address changes. Otherwise, HIP Base Exchanges from new peers could fail because they try to contact the node at an IP address it is no longer reachable at.

### 3.1 Mobility and Multi-Homing with DNS Updates

If the DNS indirectly maintains the HI->IP mapping in a FQDN->IP table, nodes can dynamically update their DNS entry in a secure fashion [5][6]. The DNS server maintaining the information will then sign and distribute the updated zone.

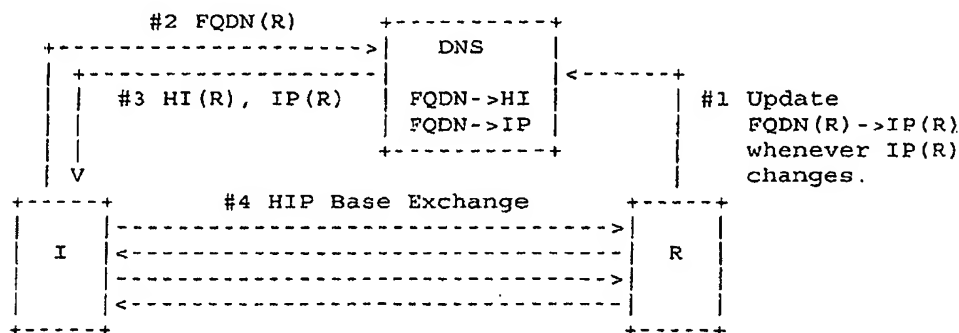


Figure 2: HIP Lookup and Base Exchange with DNS Updates

Figure 2 shows an example of this scenario. In step #1, R registers its FQDN(R)->IP(R) entry in the DNS. It will dynamically update the DNS entry whenever its IP addresses IP(R) change. Because the DNS always contains R's current IP addresses, node I can perform a HIP Base Exchange with R at its new IP address (steps #2-4).

Eggert

Expires August 5, 2004

[Page 7]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

One drawback of using dynamic DNS updates in this way is the cost of updating secure zones. Re-signing an entire zone whenever the IP addresses of one entry change places a high cost on the DNS server. Using dynamic DNS to update HI->IP mappings may thus not be appropriate when changes of IP addresses are frequent.

A simple, operational change could help limit the costs of frequent DNS updates. Instead of recomputing a zone after each dynamic update, a DNS server could aggregate the modifications and only perform zone updates periodically. The disadvantage of this approach is that HIP nodes may be unreachable until the DNS server distributes the updated zone.

Another concern with using the DNS to support HIP node mobility is the propagation time of updated DNS entries. DNS servers frequently



cache DNS responses to reduce the load on the primary servers. During the time-to-live associated with a DNS response, DNS servers may answer additional requests for the same DNS entry from their local caches instead of contacting the primary servers. Thus, even after a HIP node updates its DNS entry, the DNS can still serve the old entry until the cached responses expire. This can lead to communication problems, because peers may try to contact a HIP node at an IP address it is no longer reachable at.

### 3.2 Mobility and Multi-Homing with Rendezvous Servers

The HIP architecture tries to greatly reduce the frequency of Dynamic DNS updates by introducing Rendezvous Servers [2]. Instead of registering its current set of IP addresses in its HI->IP entry in the DNS, a HIP node may instead register the IP addresses of its Rendezvous Servers. Because the IP addresses of Rendezvous Servers are assumed to change only infrequently, this approach can significantly reduce the load on DNS servers.

Rendezvous Servers maintain a mapping between the Host Identities of HIP nodes for which they provide service and the node's current IP addresses. HIP nodes must notify their Rendezvous Servers about any changes in their IP addresses. This approach effectively relocates the HI->IP information - and the burden of keeping it current - from the DNS to the Rendezvous Servers. This can reduce update costs under the assumption that Rendezvous Servers provide more efficient ways of maintaining HI->IP tables.

When a packet destined for one of its HIP nodes arrives at a Rendezvous Server, it relays the packet to one of the HIP node's current IP addresses. Due to the specifics of the HIP, only the first packet of a HIP Base Exchange will require such relaying [2]. Subsequent packet of the HIP Base Exchange and all further data

Eggert

Expires August 5, 2004

[Page 8]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

packets will directly flow between the HIP nodes, bypassing the Rendezvous Server.

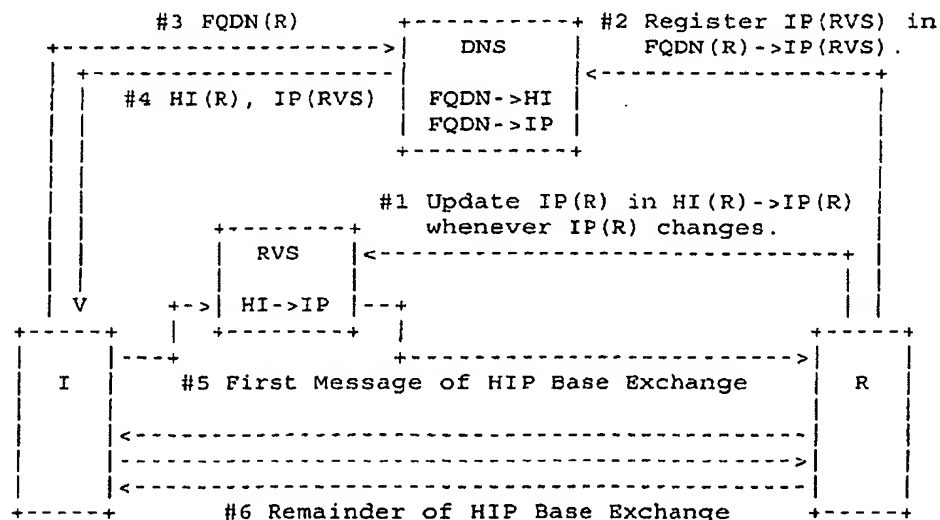


Figure 3: HIP Lookup and Base Exchange with Rendezvous Server

Figure 3 shows a HIP lookup and Base Exchange involving a Rendezvous

Server. Here, HIP node R is using Rendezvous Server RVS. In step #1, it updates RVS with its current IP addresses IP(R). Then, in step #2, R registers the Rendezvous Server's IP addresses IP(RVS) in its FQDN(R)->IP(RVS) DNS entry.

In step #3, a second HIP node I issues a DNS lookup on FQDN(R) to obtain R's Host Identities HI(R) and IP addresses. The lookup returns R's Host Identities HI(R) in step #4. The DNS reply also includes the IP addresses of the Rendezvous Server IP(RVS) (instead of IP(R), because R's current addresses are unknown to the DNS.)

In step #5, node I initiates the HIP Base Exchange. It addresses the first packet of the HIP Base Exchange to IP(RVS). Upon receipt, the Rendezvous Server relays the packet to one of R's current IP addresses IP(R). The remainder of the HIP Base Exchange then occurs directly between I and R in step #6.

When Rendezvous Servers maintain the HI->IP information, they may support more efficient update operations compared to dynamic DNS updates (Section 3.1). Unlike the DNS, Rendezvous Servers do not provide a lookup service. Instead, they use the HI->IP information to

Eggert

Expires August 5, 2004

[Page 9]

II

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

actively relay traffic between HIP nodes.

This approach changes the role of the IP addresses stored in a DNS entry. Traditionally, nodes were directly reachable at the IP addresses listed in their DNS entry. HIP Rendezvous Server change this basic property by replacing the IP addresses of their client nodes in the DNS with their own. The IP addresses in a DNS entry hence no longer directly designate interfaces of an endpoint. Instead, they identify interfaces of a node that can relay packets to the endpoint.

When two HIP nodes communicate, this change has few consequences. HIP decouples higher layers from underlying IP addresses. However, when HIP and non-HIP nodes communicate, this change has a significant impact on the overall architecture. Section 4 will discuss the implications in detail.

Eggert

Expires August 5, 2004

[Page 10]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

#### 4. Communication Between HIP and Non-HIP Nodes

Section 2 and Section 3 have discussed communication between HIP nodes. This section focuses on communication between HIP and non-HIP nodes. Two different scenarios exist. First, a HIP initiator may start communication with a non-HIP recipient. Second, a non-HIP initiator may try to contact a HIP recipient.

Without Rendezvous Servers, communication between HIP and non-HIP nodes remains identical to the current Internet. Transport-layer protocols bind directly to IP addresses. When IP addresses change, due to mobility or other reasons, transport-layer connections break.

Rendezvous Servers may establish some of HIP's benefits even if one of the endpoints does not support it. Rendezvous Servers live at static IP addresses. They can maintain ongoing transport-layer connections by acting as a relays for HIP nodes whose IP addresses may change. The discussion in the remainder of this section assumes that HIP nodes utilize Rendezvous Servers to maintain the HI->IP information as described in Section 3.

The HIP architecture document [2] discusses the role of Rendezvous Servers in HIP communication. However, it does not currently describe the details of how Rendezvous Server relay traffic between HIP and non-HIP nodes. The remainder of this section presents this aspect of Rendezvous Servers.

##### 4.1 Non-HIP Initiator to HIP Responder

In the first scenario, a non-HIP initiator starts communication with a HIP node. The HIP node is using Rendezvous Servers. Figure 4 shows this case.

Steps #1-4 remain unchanged from the HIP-HIP case shown in Figure 3 and discussed in Section 3.2. HIP node R registers the IP addresses of its Rendezvous Server RVS in the DNS. It also keeps RVS updated with its current IP addresses IP(R).

When non-HIP node I starts communication with R, it performs a DNS lookup on FQDN(R) and receives HI(R) and IP(RVS) in return. Since I does not support HIP, it disregards the Host Identity HI(R) returned by the DNS lookup. Instead, it sets up transport-layer connections using the IP addresses IP(RVS) obtained from the DNS. The Rendezvous Server RVS must then transparently relay the communication to one of R's current IP addresses IP(R) in step #5.

Eggert

Expires August 5, 2004

[Page 11]

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

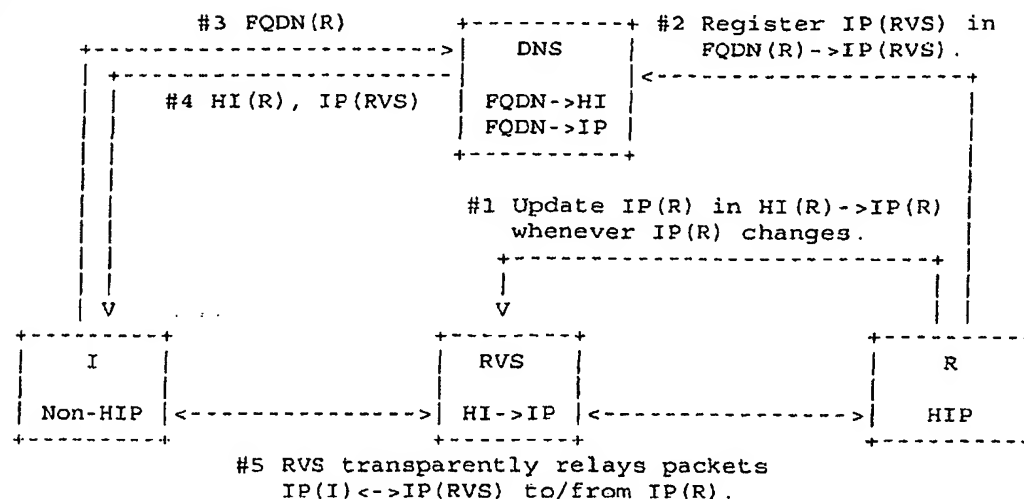


Figure 4: Non-HIP initiator to HIP responder via Rendezvous Server

End-to-end communication between I and R is complicated by the fact that R's DNS entry lists IP addresses IP(RVS). The addresses IP(RVS) belong to the Rendezvous Server RVS and not R, the endpoint of the communication. I's transport layer will thus bind connections to R to IP addresses IP(I) and IP(RVS). Section 4.3 will discuss the implications.

#### 4.2 HIP Initiator to Non-HIP Responder

This section describes a second scenario, where a HIP node initiates communication with a non-HIP node. Figure 5 shows this case.

As before, the HIP node I keeps its Rendezvous server RVS updated about its current IP addresses IP(I) in step #2. It also registers the IP addresses of the Rendezvous Server IP(RVS) in its DNS entry in step #2, instead of its own.

In step #3, I initiates a transport-layer connection to R by performing a domain name lookup on FQDN(R). The DNS reply in step #4 contains R's IP addresses IP(R) but no Host Identities, because R is not a HIP node.

Eggert

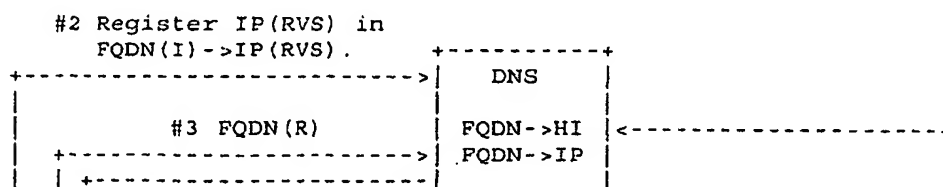
Expires August 5, 2004

[Page 12]

Internet-Draft

HIP Rendezvous Mechanisms

February 2004



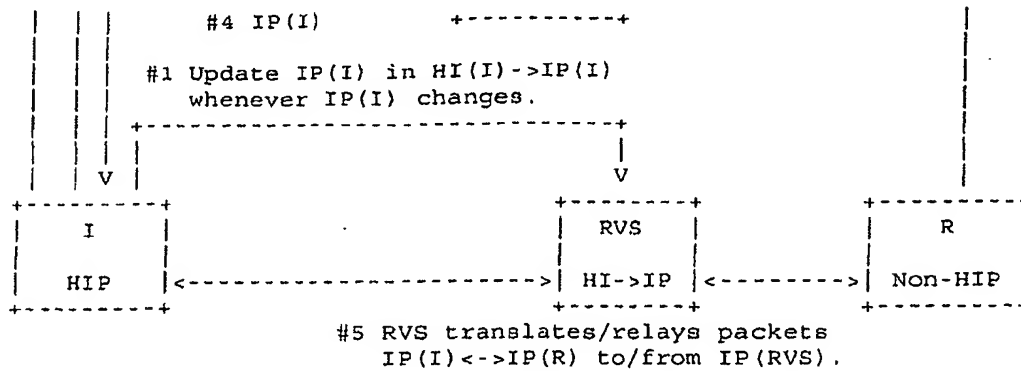


Figure 5: HIP initiator to Non-HIP responder via Rendezvous Server

If I uses IP(R) to establish a direct transport-layer connection with R, the connection will break when R's IP addresses change. Instead, R relays its traffic through Rendezvous Server RVS in step #5. Since the IP addresses of RVS are static, the transport-layer connection between I and R remains unaffected from changes to R's IP addresses.

#### 4.3 Discussion

As illustrated by the two scenarios described in Section 4.1 and Section 4.2, Rendezvous Servers can isolate non-HIP nodes from changes to their HIP peers' IP addresses. Binding transport-layer connections to static IP addresses of Rendezvous Servers, instead of the more volatile addresses of HIP peers, allows connections between HIP and non-HIP nodes to retain some of the benefits of HIP-HIP connections.

The current HIP architecture document [2] requires HIP nodes using Rendezvous Servers to register the Rendezvous Server's IP addresses in the DNS. Consequently, Rendezvous Servers become explicit connections endpoints. This causes several challenges for end-to-end communication, as discussed in the next sections.

Eggert

Expires August 5, 2004

[Page 13]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

##### 4.3.1 Relaying Overhead

The first issue is relaying overhead. When HIP nodes communicate, Rendezvous Servers will only need to relay the first packet of a HIP Base Exchange. The remaining HIP Base Exchange packets, as well as all subsequent data packets, will flow directly between the HIP nodes.

This is not the case for communication between HIP and non-HIP nodes. A non-HIP node will bind its transport-layer connection to the IP address obtained by looking up the HIP peer's domain name in the DNS. This will be the address of the Rendezvous Server.

Consequently, all data from the non-HIP to the HIP node will flow through the Rendezvous Server. This can cause significant relaying overhead. It can also increase the communication delay between the nodes, further affecting performance.

Relaying overhead will be difficult to eliminate. In order to provide

some of the benefits of HIP, non-HIP peers communicating with HIP nodes must be able to bind their transport-layer connections to static IP addresses. This constraint implies the presence of a statically addressed relay somewhere in the system.

#### 4.3.2 Return Traffic

A second issue is return traffic from the HIP node to the non-HIP node. Because a non-HIP node binds its transport-layer connection to its peer's IP address, it will not accept return traffic from a different address than it is sending to. Since all traffic from the non-HIP node is addressed to the Rendezvous Server, the non-HIP node will expect to receive return traffic from that source address.

Several approaches may address this issue. First, the HIP node may relay all its return traffic through the Rendezvous Server as well. This causes additional relaying overhead. Second, the HIP node may spoof the IP address of the Rendezvous Server when sending return traffic. This may cause problems when firewalls along the path perform ingress filtering [7]. Finally, the approach described in Section 5 can also eliminate this issue.

#### 4.3.3 Node Identification

A third issue is identification of the specific HIP node that a Rendezvous Server must relay arriving packets to. Packets arriving from non-HIP nodes are simple IP packets addressed to the Rendezvous Server. They do not contain Host Identities or other information that will allow the Rendezvous Server to identify the correct HIP node for

Eggert

Expires August 5, 2004

[Page 14]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

relaying.

One solution has the Rendezvous Server use multiple IP addresses. Each of the HIP nodes for which it provides service receives one unique IP address. The HIP node will then register this unique IP address in the DNS. Hence, the Rendezvous Server can use the destination IP addresses of arriving packets to identify the HIP node to which they must be relayed to. The approach described in Section 5 uses a similar scheme.

A downside of registering unique IP addresses per node is a more complex protocol between Rendezvous Servers and its HIP nodes. Furthermore, Rendezvous Servers serving many HIP nodes may require many IP addresses.

#### 4.3.4 Network Address Translation

The HIP architecture document [2] uses the term "forwarding" to describe the operation by which a Rendezvous Server enables the exchange of packets between communicating nodes. This document uses the term "relaying" instead, to indicate that mechanisms other than IP forwarding may suit the same purpose.

One such approach for relaying packets between HIP and non-HIP nodes is Network Address Translation [8]. When acting as a Network Address Translator, a Rendezvous Server will rewrite the IP headers of packets exchanged between communicating nodes.

The use of Network Address Translation remains problematic [9] [10]. Avoiding its use in the Rendezvous Server may improve protocol and application compatibility. Section 5 will present a rendezvous

mechanism that relays using simple IP forwarding instead, avoiding possible issues due to the use of Network Address Translation.

Eggert	Expires August 5, 2004	[Page 15]
Internet-Draft	HIP Rendezvous Mechanisms	February 2004

## 5. Rendezvous Broker

This section describes Rendezvous Brokers. Rendezvous Brokers provide a modified HIP rendezvous mechanism that addresses some of the issues discussed in Section 4.

Rendezvous Brokers are named for their similarity to tunnels brokers [11]. Rendezvous Brokers also share commonalities with MobileIP's Home Agents [12] as well as systems for leasing IP subnets [17].

Note: Rendezvous Brokers described in this section may be similar to the Packet Forwarding Agents outlined in [18]. While this similarity is under discussion, this document will use the term Rendezvous Broker for clarity. If the two concepts are deemed identical, terminology may change.

Rendezvous Brokers are IP routers and manage delegations of globally-routable IP subnets. Rendezvous Brokers may be located anywhere in the network. HIP has no concept of home networks (unlike MobileIP [12]) that would tie Rendezvous Brokers to access networks.

When a HIP node requests rendezvous service, the Rendezvous Broker delegates a unique, globally-routable IP address (or prefix) to the HIP node. HIP node and Rendezvous Broker establish a tunnel using the delegated IP address as the HIP node's tunnel endpoint address. The Rendezvous Broker installs a route towards the delegated IP address via the tunnel. At the end of this process, the HIP node is globally reachable by non-HIP nodes at the delegated IP address obtained from the Rendezvous Broker.

Figure 6 illustrates this process. In step #1, HIP node R registers its Host Identity HI(R) with the Rendezvous Broker RVB. In step #2, R receives an IP address IP(T-R) from RVB. This IP address is globally-routable and delegated to RVB.

The Rendezvous Broker and the HIP node R then establish a tunnel between themselves in step #3. IP(T-R) is the IP address of R's tunnel endpoint, T-RVB the endpoint address of the Rendezvous Broker. The tunnel encapsulates packets with IP(RVB) and IP(R). RVB then installs a route that forwards packets addressed to IP(T-R) over the tunnel.

In step #4, R registers the IP address obtained from RVB in its DNS

entry. When the non-HIP initiator I performs a DNS lookup in step #6, it receives IP(T-R) from the DNS (along with HI(R), which it ignores). I then initiates a transport-layer connection from IP(I) to IP(T-R). Packets to IP(T-R) will be routed to the RVB, because it is the router for the subnet out of which IP(T-R) was allocated. The RVB

Eggert

Expires August 5, 2004

[Page 16]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

will then forward such packets over the tunnel to R due to the route installed in step #3.

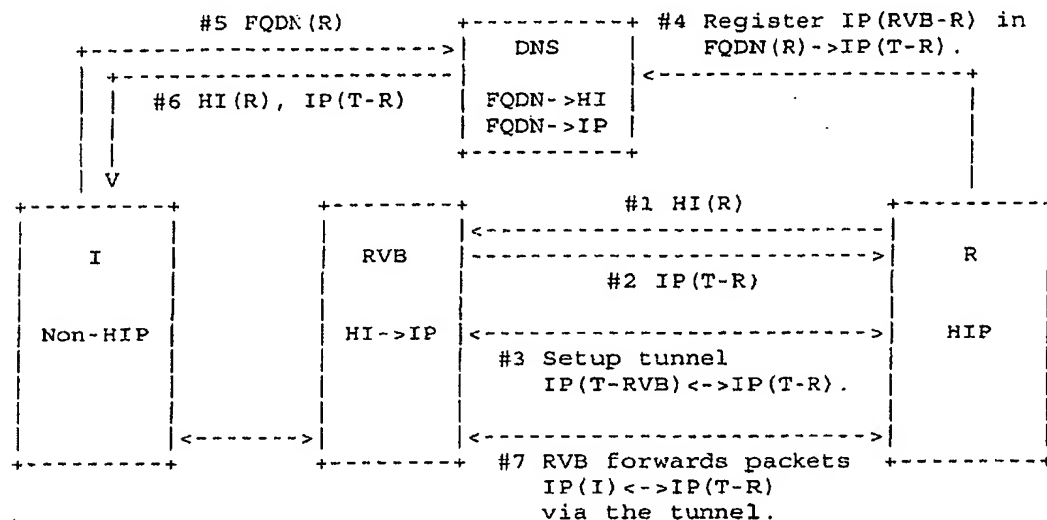


Figure 6: Non-HIP initiator to HIP responder via Rendezvous Broker

The next sections will compare Rendezvous Brokers to Rendezvous Servers and discuss several aspects of Rendezvous Brokers in more detail.

### 5.1 Comparison to Rendezvous Servers

Rendezvous Brokers address some of the shortcomings of Rendezvous Servers raised in Section 4.3. One difference is that the IP addresses in a HIP node's DNS entry again identify interfaces of the HIP node itself. With Rendezvous Servers, the DNS entry instead identifies interfaces of the Rendezvous Server.

This simplifies the operation of the Rendezvous Broker. It performs simple IP forwarding of packets that already carry the addresses of their final source and destination endpoints. Network Address Translation, or other schemes that relay by modifying packet headers, are not required. This may improve application and protocol compatibility.

Because Rendezvous Brokers are IP routers, additional mechanisms to identify the correct HIP destination node for arriving packets are not required. The globally-routable destination IP address already

Eggert

Expires August 5, 2004

[Page 17]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004



acts as a unique indicator of the final destination.

## 5.2 Mobility

Rendezvous Brokers offer mobility support that is equivalent to Rendezvous Servers. HIP nodes already notify their Rendezvous Servers when their IP addresses change. Rendezvous Brokers also require such notification.

When the IP addresses of a HIP node changes, the Rendezvous Broker and the HIP node must reconfigure the tunnel between them. This reconfiguration only affects the IP addresses used for tunnel encapsulation. The addresses of the tunnel endpoints remain unchanged. Transport-layer connections bound to a HIP node's tunnel endpoint address thus remain unaffected.

HIP nodes may change Rendezvous Servers over time and they may use multiple Rendezvous Servers at the same time. The same is true for Rendezvous Brokers. Both Rendezvous Servers and Rendezvous Brokers may be located anywhere in the network; unlike MobileIP [12], HIP has no notion of home networks. By separating rendezvous mechanisms from topological locations, HIP allows nodes to choose Rendezvous Servers or Brokers based on local criteria, including network connectivity, location, or mobility.

## 5.3 Tunneling

This document does not further define the specifics of the tunneling mechanism used between a Rendezvous Broker and its HIP nodes. Possible tunneling mechanisms include [19] [20] [21] [22] [23]. Different tunneling mechanisms incur different overheads. Some may also offer better traversal of Network Address Translators or firewalls.

Similarly, the tunnel setup protocol between Rendezvous Brokers and HIP nodes is currently unspecified. Candidate tunnel management approaches include [24] [25] [26].

Rendezvous Brokers forward all traffic from non-HIP nodes to HIP nodes over tunnels. For the return traffic from HIP nodes to non-HIP nodes two options exist. First, return traffic could also flow over tunnel. Second, return traffic could flow through the base network over one of the HIP node's interfaces. The second alternative may offer increased performance due to the avoidance of triangle routing. However, firewalls that perform ingress filtering could prevent communication [7].

Another aspect of using tunnels to connect Rendezvous Brokers and their HIP nodes is reduced Maximum Transmission Units. Implementation

Eggert

Expires August 5, 2004

[Page 18]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

issues in the network stacks of end systems and routers can lead to communication problems in such scenarios [27].

Eggert

Expires August 5, 2004

[Page 19]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

## 6. Security Considerations

The security aspects of different HIP rendezvous mechanisms are currently being investigated. They will be discussed in a future revision of this document.

Eggert Expires August 5, 2004 [Page 20]  
Internet-Draft HIP Rendezvous Mechanisms February 2004

## 7. Acknowledgments

The following people have provided thoughtful and helpful suggestions that have improved this document: Marcus Brunner, Simon Schuetz, Martin Stiernerling, and Juergen Quittek.

Eggert	Expires August 5, 2004	[Page 21]
□		
Internet-Draft	HIP Rendezvous Mechanisms	February 2004

#### Normative References

- [1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [2] Moskowitz, R., "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-05 (work in progress), October 2003.
- [3] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", draft-moskowitz-hip-08 (work in progress), October 2003.
- [4] Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-nikander-hip-mm-01 (work in progress), January 2004.
- [5] Vixie, P., Thomson, S., Rekhter, Y. and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.
- [6] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.
- [7] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [8] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [9] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.
- [10] Senie, D., "Network Address Translator (NAT)-Friendly Application Design Guidelines", RFC 3235, January 2002.
- [11] Durand, A., Fasano, P., Guardini, I. and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.
- [12] Perkins, C., "IP Mobility Support for IPv4", RFC 3344, August 2002.

Eggert Expires August 5, 2004 [Page 22]  
 Internet-Draft HIP Rendezvous Mechanisms February 2004

#### Informative References

- [13] Saltzer, J., "On the Naming and Binding of Network Destinations", RFC 1498, August 1993.
- [14] Sunshine, C., "Addressing Problems in Multi-Network Systems", IEN 178, April 1981.
- [15] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [16] Klensin, J., "Role of the Domain Name System (DNS)", RFC 3467, February 2003.
- [17] Touch, J., Eggert, L. and Y. Wang, "TetherNet Anti-NAT - Secure Internet Subnet Rental System", Proc. 3rd DARPA Information Survivability Conference and Exposition (DISCEX-III) 2003, April 2003.
- [18] Nikander, P., Ylitalo, J. and J. Wall, "Integrating Security, Mobility, and Multi-Homing in a HIP Way", Proc. Network and Distributed Systems Security Symposium (NDSS) 2003, February 2003.
- [19] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.
- [20] Levkowetz, H. and S. Vaarala, "Mobile IP Traversal of Network Address Translation (NAT) Devices", RFC 3519, May 2003.
- [21] Farinacci, D., Li, T., Hanks, S., Meyer, D. and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [22] Nikander, P., "A Bound End-to-End Tunnel (BEET) mode for ESP", draft-nikander-esp-beet-mode-00 (work in progress), October 2003.
- [23] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G. and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.
- [24] Hamzeh, K., "Ascend Tunnel Management Protocol - ATMP", RFC 2107, February 1997.
- [25] Beijnum, I., "On Demand Tunneling For Multihoming", draft-van-beijnum-multi6-odt-00 (work in progress), January 2004.

Eggert Expires August 5, 2004 [Page 23]  
 Internet-Draft HIP Rendezvous Mechanisms February 2004

- [26] Touch, J., "Dynamic Internet overlay deployment and management using the X-Bone", Computer Networks Vol. 36, No. 2-3, July 2001.

[27] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.

#### Author's Address

Lars Eggert  
 NEC Network Laboratories  
 Kurfuersten-Anlage 36  
 Heidelberg 69115  
 DE

Phone: +49 6221 90511 43  
 Fax: +49 6221 90511 55  
 EMail: lars.eggert@netlab.nec.de  
 URI: <http://www.netlab.nec.de/>

Eggert

Expires August 5, 2004

[Page 24]

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

#### Appendix A. Document Revision History

Revision	Comments
00	Initial version.

Eggert Expires August 5, 2004 [Page 25]  
□  
Internet-Draft HIP Rendezvous Mechanisms February 2004

#### Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

#### Full Copyright Statement

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION

Eggert

Expires August 5, 2004

[Page 26]

□

Internet-Draft

HIP Rendezvous Mechanisms

February 2004

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.



Eggert  
□

Expires August 5, 2004

[Page 27]

Network Working Group  
Internet-Draft  
Expires: December 16, 2003

P. Nikander  
J. Arkko  
P. Jokela  
Ericsson Research Nomadic Lab  
June 17, 2003

End-Host Mobility and Multi-Homing with Host Identity Protocol  
draft-nikander-hip-mm-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 16, 2003.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document specifies end-host mobility and multi-homing mechanisms for the Host Identity Protocol.

## Table of Contents

1.	Introduction . . . . .	4
2.	Conventions used in this document . . . . .	6
3.	Usage scenarios . . . . .	7
3.1	End-host mobility . . . . .	7
3.2	Location privacy . . . . .	7
3.3	End-host multi-homing . . . . .	7
3.4	Site multi-homing . . . . .	8
3.5	Combined mobility and multi-homing . . . . .	8
3.6	Network renumbering . . . . .	8
3.7	Combined all . . . . .	8
4.	Overview of HIP mobility and multi-homing functionality . .	9
4.1	IP addresses assigned to a node . . . . .	9
4.2	Informing the peer about multiple or changed address(es) . .	9
4.3	Address verification . . . . .	10
4.4	Forwarding Agents . . . . .	10
4.4.1	Address leases from an Forwarding Agent . . . . .	11
4.4.2	Recovering from forwarding agent crashes . . . . .	12
4.5	Security Associations . . . . .	12
5.	Protocol overview . . . . .	13
5.1	Acquiring an address lease from a Forwarding Agent . . . . .	13
5.2	Renewing an address lease . . . . .	14
5.3	Readdressing and address status . . . . .	14
6.	Protocol definition . . . . .	16
6.1	Packet formats . . . . .	16
6.1.1	REA - the HIP readdress packet . . . . .	16
6.1.2	AC and ACR - the HIP Address Check and Address Check Reply . . . . .	19
6.1.3	FAQ, FAA, FAD - the HIP Forwarding Agent packets . . . . .	20
6.2	Requesting Address Leases . . . . .	21
6.3	Providing address Leases . . . . .	21
6.4	Sending REAs . . . . .	21
6.5	Handling received REAs at end hosts . . . . .	22
7.	Policy considerations . . . . .	23
8.	Security Considerations . . . . .	24
8.1	Why does the foreign agent may require a puzzle solution? . .	24
8.2	Attacker masquerading as a FA . . . . .	24
8.3	Location privacy . . . . .	25
9.	IANA Considerations . . . . .	26
10.	Acknowledgments . . . . .	27
	Normative references . . . . .	28
	Informative references . . . . .	29
	Authors' Addresses . . . . .	29
A.	Site multi-homing . . . . .	31
A.1	A host connected to a multi-homed site . . . . .	31
A.2	Transit providers with NATs . . . . .	31
B.	Implementation experiences . . . . .	32

Intellectual Property and Copyright Statements . . . . .	33
--	----

## 1. Introduction

This document specifies how the Host Identity Protocol [3] (HIP) provides simple and efficient means for nodes to communicate while being multi-homed, mobile, or simultaneously mobile and multi-homed. Additionally, HIP allows communications even when the multi-homing or mobility causes a change in the IP version that is available in the network.

More specifically, the Host Identity Protocol [3] (HIP) defines a mechanism that decouples the transport layer from the internetworking layer, and introduces a new Host Identity namespace. When a host uses HIP, the transport layer sockets and IPsec Security Associations are not bound to IP addresses but to Host Identifiers. This document specifies how the mapping from Host Identifiers to IP addresses can be extended from a static one-to-one mapping into a dynamic one-to-many mapping. This enables end-host mobility and multi-homing. Additionally, this document introduces the concept of Forwarding Agents. A Forwarding Agents provides Mobile IP Home Agent like functionality for HIP enabled mobility.

In practice, the HIP base exchange creates a pair of IPsec Security Associations (SA) between any communicating pair of HIP enabled nodes. These SAs are not bound to IP addresses but to Host Identifiers (public keys). However, the hosts must also know at least one IP address where their peers are reachable. Initially this IP address is the one used during the HIP base exchange.

Since the SAs are not bound to IP addresses, the nodes are able to receive IPsec protected HIP packets from any address. Thus, a node can change its IP address and continue to send packets to its peers. However, the peers are not able to send replies before they can reliably and securely update the sending node's address(es).

This document specifies a mechanism that allow a HIP node to update its address(es) to its peers. The address update is implemented with a new HIP packet type, the HIP Readdress (REA) packet. Due to the danger of flooding attacks (see [4]), the peer must always check the reachability of the node before it can use the new addresses. The reachability check is implemented with a pair of new HIP packet types, HIP Address Check (AC) and HIP Address Check Reply (ACR). In addition to initiating and reachability check, an AC packet may also act as an acknowledgement for a preceding REA packet.

There are a number of situations where the simple end-to-end readdressing functionality is not sufficient. These include the initial reachability of a mobile node, location privacy, end-host and site multi-homing with legacy hosts, and NAT traversal. In these

situations there is a need for some helper functionality in the network. In this document we describe mechanisms for initial reachability, multi-homing, recovering from simultaneous movements, and combining mobility and multi-homing. Some of these functions require an additional node in the network, which has been given the name of Forwarding Agent. As a special case, a Forwarding Agent can act as a lightweight Rendezvous server as defined in [3].

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [1].

### 3. Usage scenarios

In this section we briefly introduce a number of usage scenarios where the HIP mobility and multi-homing facility is useful. To understand these usage scenarios, the reader should be at least minimally familiar with the HIP protocol specification [3]. However, for the (relatively) uninitiated reader it is most important to keep in mind that in HIP the actual payload traffic is protected with ESP, and that the ESP SPI acts as an index to the right host-to-host context.

#### 3.1 End-host mobility

A mobile IP host must change its IP address according to connectivity. The change of an IP address might be needed due to a change in the advertised IPv6 prefixes on the link, a reconnected PPP link, a new DHCP lease, or an actual movement to another subnet. In order to maintain its communication context, the host must inform its peers about the new IP address.

Although HIP enables ESP and the upper layer to be independent of the internetworking layer, there still needs to be a mapping of the pseudo-IP addresses used in the upper stack (LSI and HIT) to a real IP address. Thus, from the functional point of view, it is sufficient that the peer nodes learn the new IP address. The upper layer protocols need to know about the address and connectivity change only for QoS and similar purposes. In most cases, they need not to know.

#### 3.2 Location privacy

To protect its privacy, an IP host may want to keep its actual IP address private. Since HIP insulates the upper layer from the IP address, this can be accomplished by simple address rewriting at a privacy protecting node.

Note that a mobile node may want to keep its location private. In that case, it informs its real address to the privacy protecting node and not to the actual peers.

Full location privacy falls beyond this document.

#### 3.3 End-host multi-homing

A host may have multiple interfaces, and it is desired that it can stay reachable through all of the currently available interfaces. It is expected that the set of available interfaces may change dynamically, and that there may be policies associated with the usage



of the different interfaces. For instance, the host may have a fast but low range wireless interface and a slow high range interface. The host may generally prefer to use the fast interface, but it may be available only some of the time.

Note that a host may be multi-homed and mobile simultaneously, and that a multi-homed host may want to protect the location of some of its interface while revealing the IP address of some others.

### 3.4 Site multi-homing

A host may have an interface that has multiple globally reachable IP addresses. Such a situation may be a result of the site having multiple upper Internet Service Providers, or just because the site provides all nodes with both IPv4 and IPv6 addresses. It is desirable that the host can stay reachable with all currently available globally routable addresses, independent on how they are provided.

Note that a single interface may experience site multi-homing while the host itself may have multiple interfaces.

### 3.5 Combined mobility and multi-homing

It looks likely that in the future many of the mobile nodes will be simultaneously mobile and multi-homing, i.e., have multiple mobile interfaces. Furthermore, if the interfaces use different access technology, it is fairly likely that one of the interfaces may appear stable (retain its current IP address) while some other(s) may experience mobility (undergo IP address change).

### 3.6 Network renumbering

It is expected that IPv6 networks will be renumbered much more often than most IPv4 networks are. From an end-host point of view, network renumber is similar to mobility.

### 3.7 Combined all

It is desirable that a host may simultaneously have multiple active interfaces, be mobile (on any or all of its interfaces), utilize multiple globally reachable addresses (on any or all of its interfaces), and protect its location privacy (on any or all of its interfaces).

#### 4. Overview of HIP mobility and multi-homing functionality

HIP mobility and multi-homing is fundamentally based on the HIP architecture [4], where the transport and internetworking layers are insulated from each other with the new host identity layer. In the HIP architecture, the transport layer sockets are bound to the Host Identifiers (through HIT or LSI in the case of legacy APIs), and the Host Identifiers are translated to the actual IP address.

In the base HIP protocol specification [3], it is defined how two hosts exchange their Host Identifiers and establish a pair of ESP Security Associations (SA). The ESP SAs are then used to carry the actual payload data between the two hosts, by wrapping TCP, UDP, and other upper layer headers into transport mode ESP payloads. The IP header, carrying the ESP header, uses actual IP addresses in the network.

In the base specification, hosts use the same IP addresses for nodes that were used during the base HIP exchange. This specification defines the way how IP addresses can be changed during the communication.

##### 4.1 IP addresses assigned to a node

A host can have multiple IP addresses. It may have many interfaces that are assigned IP addresses or it may have multiple addresses assigned to one interface. There may also be multiple IP addresses in function of time: the node may change its topological location in the network, or its network may change addresses.

The interfaces are logical objects. A host may claim to have any number of interfaces, as long as a single IP address does not appear to be bound to more than one interface. The purpose of the interfaces is to group the addresses into collections that are likely to experience fate sharing. For example, if the host needs to change its addresses on interface2, it is likely that both address21 and address22 will simultaneously become obsolete.

##### 4.2 Informing the peer about multiple or changed address(es)

To be able to use effectively multiple addresses assigned to the host and update addresses that change during the communication with another node, a HIP protocol packet, the REA packet (see Section 6.1.1), is specified. The logical structure created with REA packets has three levels: hosts, interfaces, and addresses. This is illustrated in Figure 1.

address11

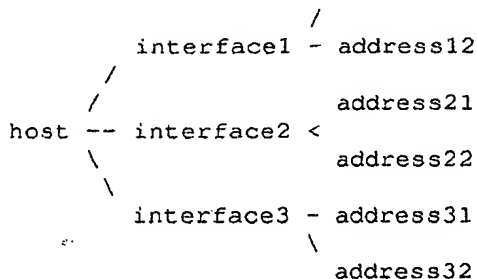


Figure 1

A single REA payload handles only one interface. To signal simultaneously changes on several interfaces, it is necessary to send several consecutive REA payloads. The packet structure supports this.

#### 4.3 Address verification

When a HIP host receives a group of IP addresses from another HIP host in a REA, it does not necessarily know for sure whether the other host is actually reachable at the claimed addresses. In fact, if the other HIP host is not fully trusted, it may be giving a bogus address with the intention of causing packet flood towards the given address [8]. Thus, before the HIP host can actually use a new address, it must first check that the peer is reachable at the new address. This is implemented with the HIP Address Check (AC) and Address Check Reply (ACR) packets.

To acknowledge that it has received the REA, and to initiate an address check, the HIP host receiving a REA immediately sends an AC to all addresses mentioned in the REA.

In a typical implementation, an address consists of the actual bitpattern used in the IP source and destination fields, a lifetime, and a status. The status is used to track the reachability of the address.

#### 4.4 Forwarding Agents

For nodes that are mobile, an IP address from where it can be reached is necessary if the node wants to be reachable by other nodes. The Forwarding Agent provides one possible solution to this. The reachability of the HIP node may require usage of both IPv6 and IPv4. If the HIP node itself is behind a network that supports only one of the IP protocol versions, the HIP node may use the FA for acting as a gateway when the peer node wants to use a IP protocol version that

the HIP node behind the FA does not directly support.

The HIP node can "lease" IP address(es) from the FA if it needs an address from where it can be reached. This can be the case, for example, when a mobile node needs a contact address for peer nodes. The HIP node contacts the FA, requests for an IP address (and SPI range), and start announcing the IP address (and some SPI) to its peers. The SPI is required if the IP address leased from the FA is not unique, i.e. it does not map one-to-one to the HIT of the HIP node. Further ESP protected data packets arriving to the FA can thus be identified using the SPI value and verifying to which HIP node that particular SPI has been reserved.

As long as the "lease" is valid, the FA will forward any packets sent to the IP address (and an SPI within the range) to the HIP host. The basic operational model is depicted in Figure 2. Without mobility (REA), using a FA results in triangular routing, as shown.

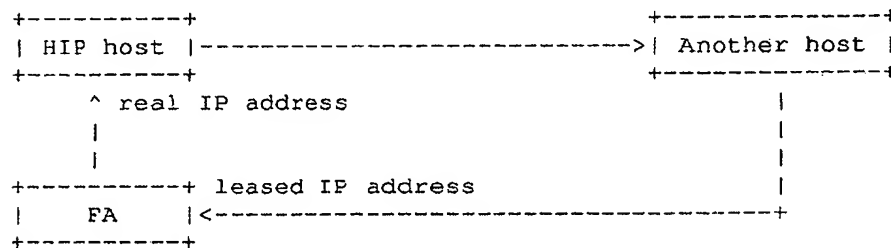


Figure 2

The actual method of discovering FAs is beyond the scope of this document, and will be specified elsewhere.

#### 4.4.1 Address leases from an Forwarding Agent

To acquire an address lease from a given FA, the HIP host sends a Forwarding Agent Query (FAQ) packet to it. In some cases, the FAQ may be sent as a broadcast or multicast packet; the details of such practice will be specified elsewhere. The HIP host may use any identity it wishes; however, the identity may be subject to local access control by the FA. That is, some FAs may be willing to serve only some HIP hosts.

If the FA is willing to serve an address to the HIP host, it replies to the FAQ with a Forwarding Agent Advice (FAA) packet. A FAA establishes an address lease to the HIP host. The HIP host can rely on the FA to keep forwarding packets to the HIP host until the address lease expires. If the FA is not willing to serve the HIP

host, it responds with a Forwarding Agent Denied (FAD) packet, specifying the reason for denial.

Each address lease has a lifetime. The HIP host may renew the address lease at any time before it the lease expires. Subject to its policy, the FA should renew and extend the lease, but it MAY refuse any extensions. In any case, it must not reduce the lease lifetime making it to expire prior to the initial expiration time.

The HIP host may abandon the lease at any time, either by failing to renew it or by sending an Forwarding Agent Query that specifies a zero lifetime. If an address lease expires without having been renewed, the FA simply discards the forwarding state and any resources associated with it.

#### 4.4.2 Recovering from forwarding agent crashes

If a FA crashes, it loses its state. Consequently, it cannot forward packets sent to it, since it does not know the IP address associated with the leased address (and the SPI). The only thing the forwarding agent could do would be to simulate lost state by the actual HIP host that is leasing the address. However, since the crashed FA does not know the HIT of the host, it cannot do that. Effectively, the forwarding agent becomes a black hole until the HIP host recognizes the situation and acquires a new lease.

It is currently an open problem whether a crashed FA should send some kind of error message to the hosts sending ESP packets to it.

#### 4.5 Security Associations

The security associations between the nodes are not any longer directly connected to the IP address of the node. The SA is associated with the HIT and there may be either one SA between the nodes, or multiple SAs when the interface capabilities require such an arrangement.

All addresses on a single interface share an SA. Multiple interfaces may share a single SA, but each interface may also have its own SA. In practice, multiple interfaces may share an SA if the experienced latency is fairly similar, in which case the packets are received within the IPsec reordering window. However, the latencies between two interfaces are considerably different, it becomes more likely that some of the packets would be discarded due to appearing to have been received outside of the ESP reordering window. Thus, in that case it is necessary to use different SAs for different interfaces.

## 5. Protocol overview

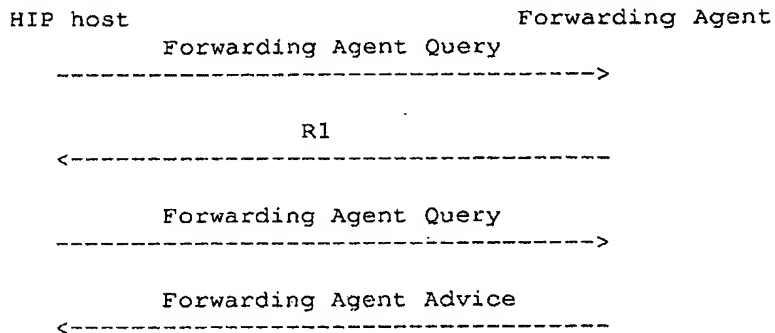
The HIP mobility and multi-homing functionality consist of the following subprotocols:

Acquiring an address lease from a Forwarding Agent

Renewing an address lease

Informing peers about multiple addresses or address changes, and verifying the reachability of addresses

### 5.1 Acquiring an address lease from a Forwarding Agent



To acquire an address lease, the HIP host sends an FAQ, requesting for an address lease. The host may specify the type of address it wants to have (IPv4, IPv6, either), the number of SPIs requested, and the desired lifetime. Any of these fields may be left empty, as well. The FAQ is always signed.

If the Forwarding agent does not trust the HIP host, it may answer with an R1, basically asking the HIP host to solve a puzzle before the Forwarding Agent is even willing to consider the request. Once the HIP host has solved the puzzle, it may send the FAQ again, this time including an answer to the puzzle.

XXX: Is it OK that the FA answers with a normal R1, or should we use some other packet type, e.g., Forwarding Agent R1 (FAR1)?

If the FA is willing to serve the HIP host, it answers with an FAA, specifying the leased IP address, its lifetime, and if the address is an IPv4 address, a range of SPIs that has been reserved for the HIP host.

## 5.2 Renewing an address lease

Once an address lease is about to expire, the HIP host may want to renew it.

HIP host

Forwarding Agent

### Forwarding Agent Query

> \*\*\*\*

### Forwarding Agent Advice

[illegible]

To renew a lease, the HIP host simply sends a FAQ specifying an existing lease, together with the desired extended lease time, and the FA replies with an FAA.

### 5.3 Readdressing and address status

HIP host

Another host

REA

[illegible]

AC

<
 [1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[7](#)
[8](#)
[9](#)
[10](#)
[11](#)
[12](#)
[13](#)
[14](#)
[15](#)
[16](#)
[17](#)
[18](#)
[19](#)
[20](#)
[21](#)
[22](#)
[23](#)
[24](#)
[25](#)
[26](#)
[27](#)
[28](#)
[29](#)
[30](#)
[31](#)
[32](#)
[33](#)
[34](#)
[35](#)
[36](#)
[37](#)
[38](#)
[39](#)
[40](#)
[41](#)
[42](#)
[43](#)
[44](#)
[45](#)
[46](#)
[47](#)
[48](#)
[49](#)
[50](#)

ACR

[illegible]

When the host changes its IP address, gets more addresses or loses an address, it may want to tell this change to peer nodes. The address changing host sends the readdress packet (REA) to the peer where it tells all available addresses. The address update packet is signed providing proof about the sending party.

As the node receiving a REA packet cannot be sure if all the received addresses are valid even the signature is correct, it sends the Address Check (AC) packet to each of the new addresses received. If the host receiving an AC message has sent the REA message that matches the received AC message, it MUST send an Address Check Reply (ACR) packet back to the peer for confirmation. New addresses received in the REA packet are ready to be used after the ACR packet has been received.

If a node receives an AC packet that does not match to any REA packet that it has sent out, it MUST drop the packet. Such a packet may be an indication that someone else is on purpose or on mistake sending a

false address to its peer.



## 6. Protocol definition

The location information update procedure in the HIP consists of the readdress packet telling the current set of addresses that the node is using, address check and address check replies. With this set of messages the IP addresses can be updated to the peer and the peer is able to verify that the addresses are valid.

In addition to the actual address update, the HIP node is provided a possibility to get leased addresses from the FA. The FA can provide address(es) (and a range of SPIs) for the HIP node and the HIP node can use this towards other nodes. The FA thus forwards packets to the HIP node when it receives packets sent to the leased address.

### 6.1 Packet formats

#### 6.1.1 REA - the HIP readdress packet

A HIP readdressing packet consists of one or more of REA\_INFO payloads, followed by a signature (HIP\_SIGNATURE) and a HMAC. The purpose of the signature is to allow middleboxes to verify the integrity of the packet. The HMAC allows the peer node to verify the packet very fast.

Intermediate systems that use the SPI will have to inspect ALL HIP packets for a REA packet. This is a potential DoS attack against the Intermediate system, as the signature processing may be relatively expensive. A further step against attack for the Intermediate systems is to implement ESP's replay protection of windowing the sequence number. This requires the intermediate system to track ALL ESP packets to follow the Sequence Number.

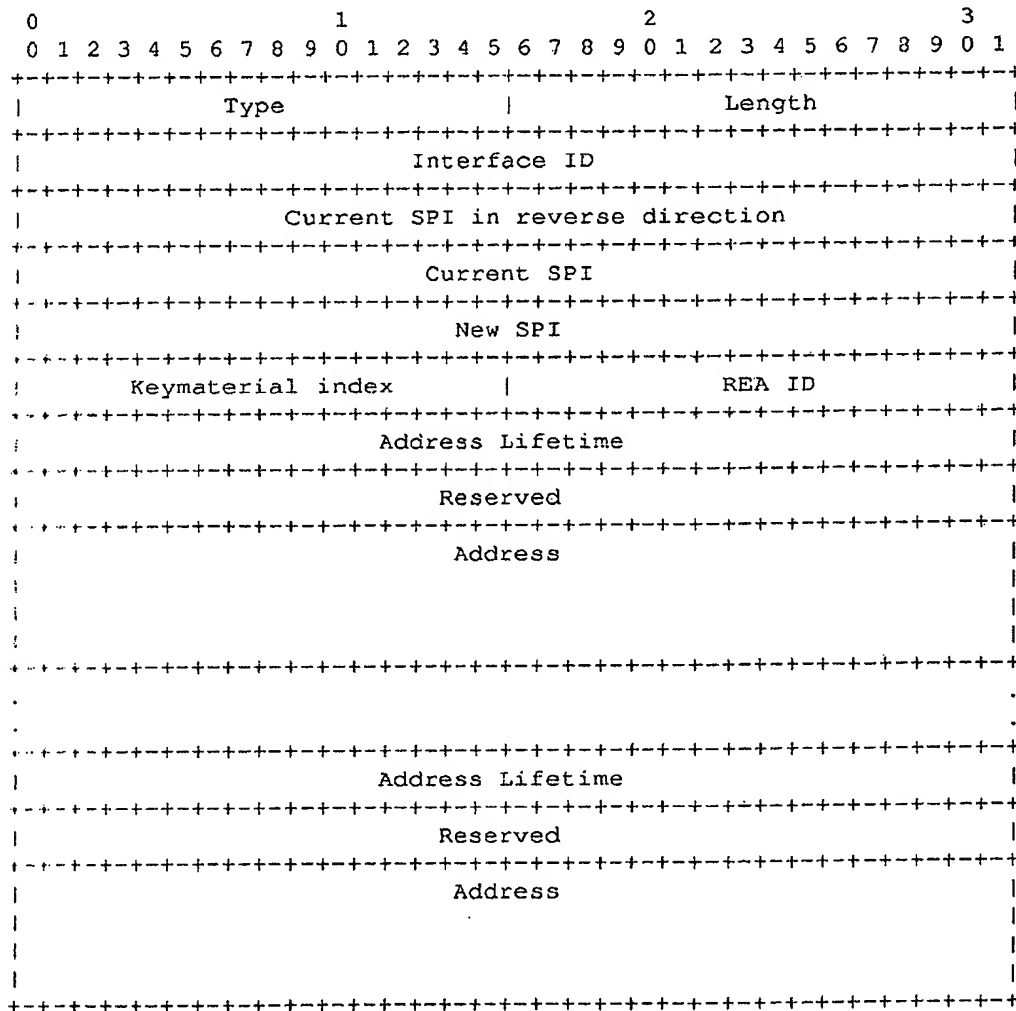
Optionally, the message may contain an ESP protected datagram. This datagram is processed as if it had arrived separately. The purpose of allowing datagrams to be embedded inside the REA packet is to increase the efficiency of transmitting the first data packet, as only one IPv6 header is required.

XXX Note (by Jari Arkko): I don't believe that this is a significant function: In fact, header compression on links is probably more efficient than this (the effect could be negative), and there might be some problems that this causes. I don't think it causes the same type of problems that piggybacking caused in Mobile IPv6, however, because the packet is always encrypted, hence it could not receive different treatment at the firewalls etc. But I'm not 100% sure that there are no other problems.

## 6.1.1.1 REA\_INFO payload

Note that the REA\_INFO payload is a kind of "expanded" NES.

XXX (Pekka): Note that I have, for the time being, removed the old ESP sequence number. Firstly, it may be hard to acquire reliably in some implementations (ours included). Secondly, we now have a REA ID field, which is basically a REA sequence number.



Type	128
Length	Length in octets, excluding Type and Length fields
Interface ID	Interface ID, as defined by the sending host
Current SPI rev.	The current SPI used in the reverse direction
Current SPI	The current SPI used for receiving ESP on this interface
New SPI	The new SPI used for receiving ESP on this interface
Keymaterial index	A bit index to the KEYMAT, where to pick up the keying material for the new SA.
REA ID	A 16-bit sequence number of nonce, used to match the REA packet to the corresponding AC packet.
Address Lifetime	Address lifetime
Reserved	Zero when sent, ignored when received
Address	An IPv6 address or an IPv4-in-IPv6 format IPv4 address

[2]

The Interface ID field identifies the (logical) interface that this packet applies to. It is implicitly qualified by the Host Identity of the sending host. The Interface ID groups a set of addresses to an interface. The purpose of the Interface ID is to allow a knowledgeable peer to interact with the sender. For example, the sender could be informing its peer that that an interface has both an IPv4 address and one or more IPv6 addresses.

The sending host is free to introduce interface IDs at will. That is, if a received REA has a new interface ID, it means that all the old addresses, assigned to other interface IDs, are also supposed to still work, while the new addresses in the REA are supposed to be associated with a new interface. On the other hand, if a received REA has an interface ID that the receiver already knows about, it would replace (all) the address(es) currently associated with the interface with the new one(s).

If the SA is changed, and if the SA is not used at any other interface any more, it MUST NOT be deleted until all ESP packets with a lower Sequence Number have been received and processed, or a reasonable time has elapsed (to account for lost packets).

#### 6.1.1.2 HMAC

The HMAC SHA-1 is used to verify a received packet.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

```

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|                                         Type                                         |
|                                         |                                         |
|                                         Length                                        |
|                                         |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
/                                         HMAC data                                         /
/                                         /
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type            65532  
Length          Length in octets, excluding Type and Length fields  
HMAC data      20 bytes of HMAC SHA-1 data

#### 6.1.2 AC and ACR - the HIP Address Check and Address Check Reply

The HIP Address Check (AC) and Address Check Reply (ACR) packets contain an AC\_INFO payload, followed by a HMAC.

In addition to acting as an address probe, the Address Check packet may also acts as an implicit acknowledgement to a REA packet.

##### 6.1.2.1 AC\_INFO payload

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|                                         Type                                         |
|                                         |                                         |
|                                         Length                                        |
|                                         |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|                                         AC ID                                         |
|                                         |                                         |
|                                         REA ID                                        |
|                                         |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|                                         RTT timestamp                                       |
|                                         |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|                                         Reserved                                       |
|                                         |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type            129  
Length          Length in octets, excluding Type and Length fields  
AC ID          A 16-bit sequence number of nonce, used to match the AC packet to the corresponding ACR packet.  
REA ID        A 16-bit sequence number of nonce, used to match the REA packet to the corresponding AC packet.  
RTT timestamp   A timestamp field which may be used for RTT estimation  
Reserved       Zero when sent, ignored when received

## 6.1.3 FAQ, FAA, FAD - the HIP Forwarding Agent packets

The HIP FAQ, FAA and FAD packets contain an FA\_INFO payload, and possibly other payloads. If a forwarding agent sends an R1 in response to FAQ, the second FAQ must also contain an BIRTHDAY\_COOKIE payload, containing the cookie response. The FAA, and FAD packets MUST contain a HOST\_ID or HOST\_ID\_FQDN payload. The FAA packet MAY contain a HOST\_ID or HOST\_ID\_FQDN payload.

## 6.1.3.1 FA\_INFO payload

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|          Type                         |          Length                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Request ID                   |          Lease ID                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Lease duration                 |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Reserved                      |          Addr type                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

IPv4 address:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|          Min SPI                       |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|          Max SPI                       |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          IPv4 address                  |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Reserved                      |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

IPv6 address:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|                                         |                                         |
|          IPv6 address (128 bits)       |                                         |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type	130
Length	Length in octets, excluding Type and Length fields

Request ID	A 16-bit sequence number of nonce, used to match the FAQ packet to the corresponding FAA/FAD packet.
Lease ID	A 16-bit number XXX
Lease duration	Duration of the lease
Reserved	Zero when sent, ignored when received
Addr type	Address type: 4 for IPv4 and 6 for IPv6 (TBD)

## 6.2 Requesting Address Leases

To request address leases from the FA, the HIP node sends an FAQ packet to the FA. The FAQ packet consists of the HIP header, FA\_INFO payload and a HIP\_SIGNATURE. If the FAQ packet is the second one sent from the HIP node to the FA, i.e. the FA responded to the first FAQ with an R1 packet, a BIRTHDAY\_COOKIE payload containing the cookie response must be included in the packet.

## 6.3 Providing address Leases

When the FA receives an FAQ packet from a HIP node, it may verify the signature in the packet. If the FA does not trust on the requesting node, it may generate an R1 packet containing a puzzle for the requesting HIP node.

If the FA trusts to the requesting HIP node, or the HIP node responded to the R1 packet with a new FAQ with a solved puzzle, the FA can allocate address(es) and possibly SPIs for the requesting node. The FA\_INFO payload may contain information about the requested addresses or the requested SPI ranges. If these requests can be met, the FA may allocate address(es) and possible SPIs for the requesting node.

If the allocation request was accepted and a successful reservation of data was performed, the FA responds to the requesting node with a FAA packet. The FAA consists of an FA\_INFO payload describing the address(es) and possible SPIs that are reserved for the requesting node.

However, if the FA was not able to allocate address(es), SPIs or the request was malformed, the FA responds with a FAD packet.

## 6.4 Sending REAs

The HIP node may want to send address information to the peer node whenever there are updates in the addresses that are assigned to it. The leased addresses can be considered also to be possible addresses and they must be assigned to a logical interface.

The REA packet contains the HIP header, one or more REA\_INFO, HIP\_SIGNATURE and HMAC TLVs. The REA\_INFO describes all the addresses that are associated with that particular interface identifier. If a previously associated address is not included in the list, the peer considers it as a lost address and removes it from the address associations.

#### 6.5 Handling received REAs at end hosts

When a HIP node receives a REA packet, it verifies the signature in it. If the packet was valid, it may initiate an address check procedure. The address check (AC) packet is sent to all addresses that were listed in the REA\_INFO payload. The HIP node receiving the REA packet from a node that it trusts, may accept all addresses without making any address check for them. If ACs are sent, the addresses in the REA\_INFO must not be used until corresponding ACR packet is received from the peer node.

7. Policy considerations

TBD



## 8. Security Considerations

(Initial text by Jari Arkko): If not controlled in some manner, messaging related to address changes would create the following types of vulnerabilities:

- Revealing the contents of the (cleartext) communications

- Hijacking communications and man-in-the-middle attacks

- Denial of service for the involved nodes, by disabling their ability to receive the desired communications

- Denial of service for third parties, by redirecting a large amount of traffic to them

- Revealing the location of the nodes to other parties

In HIP, communications are bound to the public keys of the end-points and not to IP addresses. The REA message is signed with the sender's private key, and hence it becomes impossible to hijack the communications of another node through the use of the REA message. Similarly, since only the node itself can sign messages to move its traffic flows to a new IP address, denial of service attacks through REA can not cause the traffic flows to be sent to an IP address that the node did not wish to use. Finally, In HIP all communications are encrypted with ESP, so a hijack attempt would also be unable to reveal the contents of the communications.

Malicious nodes that use HIP can, however, try to cause a denial of service attack by establishing a high-volume traffic flow, such as a video stream, and then redirecting it to a victim. However, the return routability check provides some assurance that the given address is willing to accept the new traffic. Only attackers who are on the path between the peer and the new address could respond to the test.

### 8.1 Why does the foreign agent may require a puzzle solution?

In Section 5.1 it is stated that a foreign agent may pass a puzzle, in an R1, to the HIP host if it does not trust the HIP host. This protects the foreign agent from CPU consuming DoS attacks. If this protection were not used, an attacker could send a stream of bogus FAQ packets to the foreign agent, making it to spend all of its CPU to verify signatures that might be full garbage.

### 8.2 Attacker masquerading as a FA

The ability for an attacker to masquerade as a forwarding agent depends on how the HIP host locates forwarding agents. That, in turn, falls beyond the scope of this document. However, if the HIP host accepts services from unknown or untrusted forwarding agents, it is taking a risk of getting a black hole or eavesdropped address. The resulting attack is usually not very serious, though, since all actual data traffic is protected with ESP, and the HIP packets are signed. Thus, the worst an untrustworthy forwarding agent can do is to blackhole the packets.

### 8.3 Location privacy

TBD

## 9. IANA Considerations

## 10. Acknowledgments

Thanks to Kimmo Nieminen.

## Normative references

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.
- [3] Nikander, P. and R. Moskowitz, "Host Identity Protocol", draft-moskowitz-hip-06 (work in progress), May 2003.
- [4] Moskowitz, R., "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-03 (work in progress), May 2003.

## Informative references

- [5] Bellovin, S., "EIDs, IPsec, and HostNAT", IETF 41th, March 1998.
- [6] Nikander, P., Ylitalo, J. and J. Wall, "Integrating Security, Mobility, and Multi-Homing in a HIP Way", Network and Distributed Systems Security Symposium (NDSS'03), February 6-7, 2003, San Diego, CA, pages 87-99, Internet Society, February 2003.
- [7] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", draft-iab-sec-cons-03 (work in progress), February 2003.
- [8] Nikander, P., Aura, T., Arkko, J., Montenegro, G. and E. Nordmark, "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-00 (work in progress), April 2003.

## Authors' Addresses

Pekka Nikander  
Ericsson Research Nomadic Lab

JORVAS FIN-02420  
FINLAND

Phone: +358 9 299 1  
EMail: pekka.nikander@nomadiclab.com

Jari Arkko  
Ericsson Research Nomadic Lab

JORVAS FIN-02420  
FINLAND

Phone: +358 9 299 1  
EMail: jari.arkko@nomadiclab.com

Internet-Draft

HIP Mobility and Multi-Homing

June 2003

Petri Jokela  
Ericsson Research Nomadic Lab

JORVAS FIN-02420  
FINLAND

Phone: +358 9 299 1  
Email: petri.jokela@nomadiclab.com

Nikander, et al.

Expires December 16, 2003

[Page 30]

## Appendix A. Site multi-homing

A site, connected to multiple transit providers, is considered to be multi-homed. There is a possibility to send traffic using any of the transit provider networks. A node, connected to a multi-homed site, can experience this multi-homing from the received IP addresses.

### A.1 A host connected to a multi-homed site

When a node connects to a multi-homed network, it may receive multiple IP addresses on this connected interface. These addresses can be either local IP addresses (behind a NAT) or public addresses.

A traditional node setting up a connection, selects one of the available local addresses for this particular connection. This address cannot be changed for the existing connection.

A HIP node experiencing similar site multi-homing is not limited to the one source address selected during the connection set up. The node has multiple IP addresses on one interface and the mapping between the Host Identity - Interface - IP addresses is a mapping from one to one to many. The used IP address can be changed while the connection exists.

When configured multiple addresses to one interface, the node can update this list of addresses to peer nodes. Thus, the different transit providers can be used according to policies defined in the node. The policies can be defined locally or they can be received by other methods. (see policies, TBD)

### A.2 Transit providers with NATs

A transit provider may have NAT boxes in the network. The HIP node connected to a site that is further connected to a transit provider using NATs, must get the knowledge about the NAT box between itself and the peer node. The address that the HIP node sends to the peer node must be a public one, thus NATted address is not valid.

The NAT box on the path must support address (and SPI) leasing for the HIP node. When the HIP node finds out that there is a NAT box, the host must get a leased address (or a set of addresses) from the NAT. These addresses are routable at the other side of the NAT. They still need not to be globally routable as there may be another NAT box on the path.



Appendix B. Implementation experiences

## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

## Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF  
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the  
Internet Society.

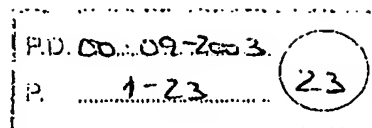
Generated by unregistered txt2pdf PRO v.6.7 © SANFACE Software 2003  
Available at <http://www.sanface.com/txt2pdf.html>

Network Working Group  
 Internet-Draft  
 Expires: March 1, 2004

R. Moskowitz  
 ICSAlabs, a Division of TruSecure  
 Corporation  
 P. Nikander  
 Ericsson Research Nomadic Lab  
 Sep 2003

XP-002300906

Host Identity Protocol Architecture  
 draft-moskowitz-hip-arch-05



#### Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 1, 2004.

#### Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

#### Abstract

This memo describes the reasoning behind a proposed new namespace, the Host Identity namespace, and a new protocol layer, the Host Identity Protocol, between the internetworking and transport layers. Herein are presented the basics of the current namespaces, strengths and weaknesses, and how a new namespace will add completeness to them. The roles of this new namespace in the protocols are defined.

Moskowitz & Nikander Expires March 1, 2004 [Page 1]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

#### Table of Contents

1.	Introduction . . . . .	3
2.	Background . . . . .	4
2.1	A Desire for a Namespace for Computing Platforms . . . . .	5
3.	Host Identity Namespace . . . . .	7
3.1	Host Identifiers . . . . .	7
3.2	Storing Host Identifiers in DNS . . . . .	8
3.3	Host Identity Tag (HIT) . . . . .	8
3.4	Local Scope Identifier (LSI) . . . . .	9
4.	New Stack Architecture . . . . .	10

4.1	Transport associations and endpoints . . . . .	10
5.	End-Host Mobility and Multi-Homing . . . . .	12
5.1	Rendezvous server . . . . .	12
5.2	Protection against Flooding Attacks . . . . .	13
6.	HIP and IPsec . . . . .	14
7.	HIP and NATs . . . . .	15
7.1	HIP and TCP Checksum . . . . .	15
8.	HIP Policies . . . . .	16
9.	Benefits of HIP . . . . .	17
9.1	HIP's Answers to NSRG questions . . . . .	18
10.	Security Considerations . . . . .	20
10.1	HITS used in ACLs . . . . .	21
10.2	Non-security Considerations . . . . .	22
11.	Acknowledgments . . . . .	23
	References (informative) . . . . .	24
	Authors' Addresses . . . . .	24
	Intellectual Property and Copyright Statements . . . . .	26

Moskowitz & Nikander Expires March 1, 2004 [Page 2]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## 1. Introduction

The Internet has created two global namespaces: Internet Protocol (IP) addresses and Domain Name Service (DNS) names. These two namespaces have a set of features and abstractions that have powered the Internet to what it is today. They also have a number of weaknesses. Basically, since they are all we have, we try and do too much with them. Semantic overloading and functionality extensions have greatly complicated these namespaces.

The Host Identity namespace fills an important gap between the IP and DNS namespaces. The Host Identity namespace consist of Host Identifiers (HI). A Host Identifier is cryptographic in its nature; it is the public key of an asymmetric key-pair. A Host Identity is assigned to each host, or technically its networking kernel or stack. Each host will have at least one Host Identity and a corresponding Host Identifier, which can either be public (e.g. published in DNS), or anonymous. Client systems will tend to have both public and anonymous Identities.

Although the Host Identities could be used in many authentication systems, the presented architecture introduces a new protocol, called the Host Identity Protocol (HIP), and a cryptographic exchange,

called the HIP base exchange [4]. The new protocol provides for limited forms of trust between systems. It enhances mobility, multi-homing and dynamic IP renumbering [7], aids in protocol translation / transition [4], and reduces certain types of denial-of-service (DoS) attacks [4].

When HIP is used, the actual payload traffic between two HIP hosts is typically protected with IPsec. The Host Identities are used to create the needed IPsec Security Associations (SA) and to authenticate the hosts. The actual payload IP packets do not differ in any way from standard IPsec protected IP packets.

Moskowitz & Nikander Expires March 1, 2004 [Page 3]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## 2. Background

The Internet is built from three principle components: computing platforms, packet transport (i.e. internetworking) infrastructure, and services (applications). The Internet exists to service two principal components: people and robotic processes (silicon based people, if you will). All these components need to be named in order to interact in a scalable manner.

There are two principal namespaces in use in the Internet for these components: IP numbers, and Domain Names. Email, HTTP and SIP addresses are really only extensions of Domain Names.

IP numbers are a confounding of two namespaces, the names of the networking interfaces and the names of the locations ('confounding' is a term used in statistics to discuss metrics that are merged into one with a gain in indexing, but a loss in informational value). The names of locations should be understood as denoting routing direction vectors, i.e., information that is used to deliver packets to their destinations.

IP numbers name networking interfaces, and typically only when the interface is connected to the network. Originally IP numbers had long-term significance. Today, the vast number of interfaces use ephemeral and/or non-unique IP numbers. That is, every time an interface is connected to the network, it is assigned an IP number.

In the current Internet, the transport layers are coupled to the IP addresses. Neither can evolve separately from the other. IPng deliberations were framed by concerns of requiring a TCPng effort as well.

Domain Names provide hierarchically assigned names for some computing platforms and some services. Each hierarchy is delegated from the level above; there is no anonymity in Domain Names.

Email addresses provide naming for both humans and autonomous applications. Email addresses are extensions of Domain Names, only in so far as a named service is responsible for managing a person's mail. There is some anonymity in Email addresses.

There are three critical deficiencies with the current namespaces. Firstly, dynamic readdressing cannot be directly managed. Secondly, anonymity is not provided in a consistent, trustable manner. Finally, authentication for systems and datagrams is not provided. All because computing platforms are not well named with the current namespaces.

Moskowitz & Nikander Expires March 1, 2004

[Page 4]

□

Internet-Draft Host Identity Protocol Architecture

Sep 2003

## 2.1 A Desire for a Namespace for Computing Platforms

An independent namespace for computing platforms could be used in end-to-end operations independent of the evolution of the internetworking layer and across the many internetworking layers. This could support rapid readdressing of the internetworking layer either from mobility or renumbering.

If the namespace for computing platforms is cryptographically based, it can also provide authentication services. If this namespace is locally created without requiring registration, it can provide anonymity.

Such a namespace (for computing platforms) and the names in it should have the following characteristics:

The namespace should be applied to the IP 'kernel'. The IP kernel is the 'component' between services and the packet transport infrastructure.

The namespace should fully decouple the internetworking layer from the higher layers. The names should replace all occurrences of IP addresses within applications (like in the TCB). This may require changes to the current APIs. In the long run, it is probable that some new APIs are needed.

The introduction of the namespace should not mandate any administrative infrastructure. Deployment must come from the bottom up, in a pairwise deployment.

The names should have a fixed length representation, for easy inclusion in datagrams and programming interfaces (e.g the TCB).

Using the namespace should be affordable when used in protocols. This is primarily a packet size issue. There is also a computational concern in affordability.

The names must be statistically globally unique. 64 bits is inadequate (1% chance of collision in a population of 640M); thus approximately 100 or more bits should be used.

The names should have a localized abstraction so that it can be used in existing protocols and APIs.

It must be possible to create names locally. This can provide anonymity at the cost of making resolvability very difficult.



Sometimes the names may contain a delegation component. This is

Moskowitz & Nikander Expires March 1, 2004 [Page 5]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

the cost of resolvability.

The namespace should provide authentication services. This is a preferred function.

The names should be long lived, but replaceable at any time. This impacts access control lists; short lifetimes will tend to result in tedious list maintenance or require a namespace infrastructure for central control of access lists.

In this document, such a new namespace is called the Host Identity namespace. Using Host Identities requires its own protocol layer, the Host Identity Protocol, between the internetworking and transport layers. The names are based on public key cryptography to supply authentication services. Properly designed, it can deliver all of the above stated requirements.

Moskowitz & Nikander Expires March 1, 2004 [Page 6]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

### 3. Host Identity Namespace

A name in the Host Identity namespace, a Host Identifier (HI),

represents a statistically globally unique name for naming any system with an IP stack. This identity is normally associated, but not limited to, an IP stack. A system can have multiple identities, some 'well known', some anonymous. A system may self assert its identity, or may use a third-party authenticator like DNSSEC, PGP, or X.509 to 'notarize' the identity assertion. It is expected that the Host Identifiers will initially be authenticated with DNSSEC and that all implementations will support DNSSEC as a minimal baseline.

There is a subtle but important difference between Host Identities and Host Identifiers. An Identity refers to the abstract entity that is identified. An Identifier, on the other hand, refers to the concrete bit pattern that is used in the identification process.

In theory, any name that can claim to be 'statistically globally unique' may serve as a Host Identifier. However, in the authors' opinion, a public key of a 'public key pair' makes the best Host Identifiers. As documented in the Host Identity Protocol specification [4], a public key based HI can authenticate the HIP packets and protect them for man-in-the-middle attacks. Since authenticated datagrams are mandatory to provide much of HIP's denial-of-service protection, the Diffie-Hellman exchange in HIP has to be authenticated. Thus, only public key HI and authenticated HIP messages are supported in practice. In this document, the non-cryptographic forms of HI and HIP are presented to complete the theory of HI, but they should not be implemented as they could produce worse denial-of-service attacks than the Internet has without Host Identity.

### 3.1 Host Identifiers

Host Identity adds two main features to Internet protocols. The first is a decoupling of the internetworking and transport layers; see Section 4. This decoupling will allow for independent evolution of the two layers. Additionally, it can provide end-to-end services over multiple internetworking realms. The second feature is host authentication. Because the Host Identifier is a public key, this key can be used to authenticate security protocols like IPsec.

The only completely defined structure of the Host Identity is that of a public key pair. In this case, the Host Identity is referred to by its public component, the public key. Thus, the name representing a Host Identity in the Host Identity namespace, i.e. the Host Identifier, is the public key. In a way, the possession of the private key defines the Identity itself. If the private key is

Moskowitz & Nikander	Expires March 1, 2004	[Page 7]
□ Internet-Draft	Host Identity Protocol Architecture	Sep 2003

possessed by more than one node, the Identity can be considered to be a distributed one.

Architecturally, any other Internet naming convention might form a usable base for Host Identifiers. However, non-cryptographic names should only be used in situations of high trust - low risk. That is any place where host authentication is not needed (no risk of host spoofing) and no use of IPsec. The current HIP documents do not specify how to use any other types of Host Identifiers but public keys.

The actual Host Identities are never directly used in any Internet protocols. The corresponding Host Identifiers (public keys) may be stored in various DNS or LDAP directories as identified elsewhere in this document, and they are passed in the HIP base exchange. A Host

Identity Tag (HIT) is used in other protocols to represent the Host Identities. Another representation of the Host Identities, the Local Scope Identifier (LSI), can also be used in protocols and APIs.

### 3.2 Storing Host Identifiers in DNS

The Host Identifiers should be stored in DNS. The exception to this is anonymous identities. The HI is stored in a new RR type, to be defined. This RR type is likely to be quite similar to the IPSECKEY RR [5].

Alternatively, or in addition to storing Host Identifiers in the DNS, they may be stored in various kinds of Public Key Infrastructure (PKI). Such a practice may allow them to be used for purposes other than pure host identification.

### 3.3 Host Identity Tag (HIT)

A Host Identity Tag is an 128-bit representation for a Host Identity. It is created by taking a cryptographic hash over the corresponding Host Identifier. There are two advantages of using a hash over using the Host Identifier in protocols. Firstly, its fixed length makes for easier protocol coding and also better manages the packet size cost of this technology. Secondly, it presents the identity in a consistent format to the protocol independent of the whatever underlying technology is used.

In the HIP packets, the HITs identify the sender and recipient of a packet. Consequently, a HIT should be unique in the whole IP universe. In the extremely rare case that a single HIT happens to map to more than one Host Identities, the Host Identifiers (public keys) will make the final difference. If there is more than one public key for a given node, the HIT acts as a hint for the correct

Moskowitz & Nikander	Expires March 1, 2004	[Page 8]
□		
Internet-Draft	Host Identity Protocol Architecture	Sep 2003

public key to use.

### 3.4 Local Scope Identifier (LSI)

An LSI is a 32-bit localized representation for a Host Identity. The purpose of an LSI is to facilitate using Host Identities in existing protocols and APIs. LSI's advantage over HIT is its size; its disadvantage is its local scope. The generation of LSIs is defined in the Host Identity Protocol specification [4].

Examples of how LSIs can be used include: as the address in a FTP command and as the address in a socket call. Thus, LSIs act as a bridge for Host Identities into old protocols and APIs.

Moskowitz & Nikander Expires March 1, 2004 [Page 9]  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

#### 1. New Stack Architecture

One way to characterize Host Identity is to compare the proposed new architecture with the current one. As discussed above, the IP addresses can be seen to be a confounding of routing direction vectors and interface names. Using the terminology from the IRTF Name Space Research Group Report [6] and, e.g., the unpublished Internet-Draft Endpoints and Endpoint Names [9] by Noel Chiappa, the IP addresses currently embody the dual role of locators and endpoint identifiers. That is, each IP address names a topological location in the Internet, thereby acting as a routing direction vector, or locator. At the same time, the IP address names the physical network interface currently located at the point-of-attachment, thereby acting as an endpoint name.

In the HIP architecture, the endpoint names and locators are separated from each other. IP addresses continue to act as locators. The Host Identifiers take the role of endpoint identifiers. It is important to understand that the endpoint names based on Host Identities are slightly different from interface names; a Host Identity can be simultaneously reachable through several interfaces.

The difference between the bindings of the logical entities are illustrated in Figure 1.

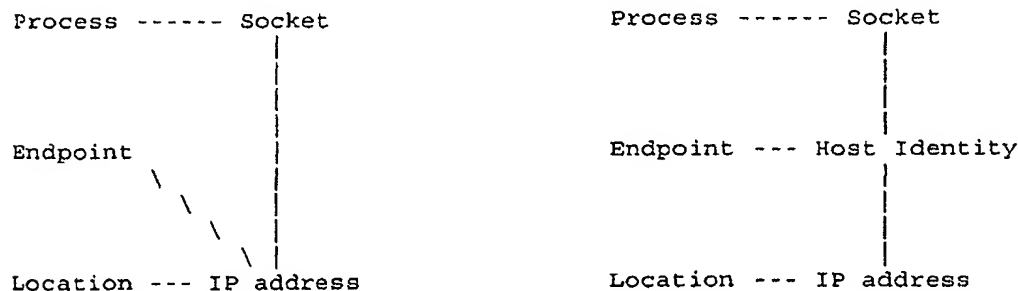


Figure 1

## 4.1 Transport associations and endpoints

Architecturally, HIP provides for a different binding of transport layer protocols. That is, the transport layer associations, i.e., TCP connections and UDP associations, are no more bound to IP addresses but to Host Identities.

Moskowitz & Nikander Expires March 1, 2004 [Page 10]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

It is possible that a single physical computer hosts several logical endpoints. With HIP, each of these endpoints would have a distinct Host Identity. Furthermore, since the transport associations are bound to Host Identities, HIP provides for process migration and clustered servers. That is, if a Host Identity is moved from one physical computer to another, it is also possible to simultaneously move all the transport associations without breaking them. Similarly, if it is possible to distribute the processing of a single Host Identity over several physical computers, HIP provides for cluster based services without any changes at the client endpoint.

HIP decouples the transport from the internetworking layer, and binds the transport associations to the Host Identities (through actually either the HIT or LSI). Consequently, HIP can provide for a degree of internetworking mobility and multi-homing at a very low infrastructure cost. HIP mobility includes IP address changes (via any method) to either party. Thus, a system is considered mobile if its IP address can change dynamically for any reason like PPP, DHCP, IPv6 prefix reassignments, or a NAT device remapping its translation. Likewise, a system is considered multi-homed if it has more than one globally routable IP address at the same time. HIP allows these IP addresses to be linked with each other, and if one address becomes unusable (e.g. due to a network failure), existing transport associations can be easily moved to another address.

When a node moves while communication is already on-going, address changes are rather straightforward. The peer of the mobile node can just accept a HIP or an integrity protected IPsec packet from any address and totally ignore the source address. However, as discussed in Section 5.2 below, a mobile node must send a HIP readdress packet to inform the peer of the new address(es), and the peer must verify that the mobile node is reachable through these addresses. This is especially helpful for those situations where the peer node is sending data periodically to the mobile node (that is re-starting a connection after the initial connection).

Making a contact to a mobile node is slightly more involved. In order to start the HIP exchange, the initiator node has to know how to reach the mobile node. Although Dynamic DNS could be used for this function for infrequently moving nodes, an alternative to using DNS in this fashion is to use a piece of new static infrastructure called a HIP rendezvous server. Instead of registering its current dynamic address to the DNS server, the mobile node registers the address(es) of its rendezvous server(s). The mobile node keeps the rendezvous server(s) continuously updated with its current IP address(es). A rendezvous server simply forwards the initial HIP packet from an initiator to the mobile node at its current location. All further packets flow between the initiator and the mobile node. There is typically very little activity on a rendezvous server, address updates and initial HIP packet forwarding. Thus, one server can support a large number of potential mobile nodes. The mobile nodes must trust the rendezvous server to properly maintain their HIT and IP address mappings.

The rendezvous server is also needed if both of the nodes are mobile

and happen to move at the same time. In that case, the HIP readdress packets will cross each other in the network and never reach the peer node. To solve this situation, the nodes should remember the rendezvous server address, and re-send the HIP readdress packet to the rendezvous server if no reply is received.

The mobile node keeps its address current on the rendezvous server by

setting up a HIP association with the rendezvous server and sending HIP readdress packets to it. A rendezvous server will permit two mobile systems to use HIP without any extraneous infrastructure (in addition to the rendezvous server itself), including DNS if they have a method other than a DNS query to get each other's HI and HIT.

## 5.2 Protection against Flooding Attacks

While the idea of informing about address changes by simply sending packets with a new source address appears appealing, it is not secure enough. That is, even if HIP does not rely on the source address for anything (once the base exchange has been completed), it appears to be necessary to check a mobile node's reachability at the new address before actually sending any larger amounts of traffic to the new address.

Blindly accepting new addresses would potentially lead to flooding Denial-of-Service attacks against third parties [8]. In a distributed flooding attack an attacker opens (anonymous) high volume HIP connections with a large number of hosts, and then claims to all of these hosts that it has moved to a target node's IP address. If the peer hosts were to simply accept the move, the result would be a packet flood to the target node's address. To close this attack, HIP includes an address check mechanism where the reachability of a node is separately checked at each address before using the address for larger amounts of traffic.

Whenever HIP is used between two hosts that fully trust each other, the hosts may optionally decide to skip the address tests. However, such performance optimization must be restricted to peers that are known to be trustworthy and capable of protecting themselves from malicious software.

Moskowitz & Nikander Expires March 1, 2004 [Page 13]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## 6. HIP and IPsec

The preferred way of implementing HIP is to use IPsec to carry the actual data traffic. As of today, the only completely defined method is to use IPsec Encapsulated Security Payload (ESP) to carry the data packets. In the future, other ways of transporting payload data may be developed, including ones that do not use cryptographic protection.

In practise, the HIP base exchange uses the cryptographic Host Identifiers to set up a pair of ESP Security Associations (SAs) to enable ESP in an end-to-end manner. This is implemented in a way that can span addressing realms.

From a conceptual point of view, the IPsec Security Parameter Index (SPI) in ESP provides a simple compression of the HITs. This does require per-HIT-pair SAs (and SPIs), and a decrease of policy granularity over other Key Management Protocols, such as IKE and IKEv2. Future HIP extensions may provide for more granularity and

creation of several ESP SAs between a pair of HITs

Since HIP is designed for host usage, not for gateways, only ESP transport mode is supported. An ESP SA pair is indexed by the SPIs and the two HITs (both HITs since a system can have more than one HIT). The SAs need not to be bound to IP addresses; all internal control of the SA is by the HITs. Thus, a host can easily change its address using Mobile IP, DHCP, PPP, or IPv6 readdressing and still maintain the SAs. Since the transports are bound to the SA (via an LSI or a HIT), any active transport is also maintained. Thus, real world conditions like loss of a PPP connection and its re-establishment or a mobile handover will not require a HIP negotiation or disruption of transport services.

Since HIP does not negotiate any SA lifetimes, all lifetimes are local policy. The only lifetimes a HIP implementation MUST support are sequence number rollover (for replay protection), and SA timeout. An SA times out if no packets are received using that SA. Implementations MAY support lifetimes for the various ESP transforms.

Moskowitz & Nikander Expires March 1, 2004 [Page 14]  
 □ Internet-Draft Host Identity Protocol Architecture Sep 2003

## 7. HIP and NATs

Passing packets between different IP addressing realms requires changing IP addresses in the packet header. This may happen, for example, when a packet is passed between the public Internet and a private address space, or between IPv4 and IPv6 networks. The address translation is usually implemented as Network Address Translation (NAT) [2] or NAT Protocol translation (NAT-PT) [1].

In a network environment where the identification is based on the IP addresses, identifying the communicating nodes is difficult when NAT is used. With HIP, the transport layer endpoints are bound to the Host Identities. Thus, a connection between two hosts can traverse many addressing realm boundaries. The IP addresses are used only for routing purposes; the IP addresses may be changed freely during packet traversal.

For a HIP based flow, a NAT or NAT-PT system tracks the mapping of HITs and the corresponding IPsec SPIs to an IP address. Many HITs can map to a single IP address on a NAT, simplifying connections on address poor NAT interfaces. The NAT can gain much of its knowledge from the HIP packets themselves; however, some NAT configuration may be necessary.

The NAT systems cannot touch the datagrams within the IPsec envelope, thus application specific address translation must be done in the end systems. HIP provides for 'Distributed NAT', and uses the HIT or the LSI as a place holder for embedded IP addresses.

## 7.1 HIP and TCP Checksum



There is no way for a host to know if any of the IP addresses in the IP header are the addresses used to calculate the TCP checksum. That is, it is not feasible to calculate the TCP checksum using the actual IP addresses in the pseudo header; the addresses received in the incoming packet are not necessarily the same as they were on the sending host. Furthermore, it is not possible to recompute the upper layer checksums in the NAT/NAT-PT system, since the traffic is IPsec protected. Consequently, the TCP and UDP checksums are calculated using the HITs in the place of the IP addresses in the pseudo header. Furthermore, only the IPv6 pseudo header format is used. This provides for IPv4 / IPv6 protocol translation.

Moskowitz & Nikander Expires March 1, 2004 [Page 15]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## 8. HIP Policies

There are a number of variables that will influence the HIP exchanges that each host must support. All HIP implementations should support at least 2 HIs, one to publish in DNS and one for anonymous usage. Although anonymous HIs will be rarely used as responder HIs, they are likely be common for initiators. Support for multiple HIs is recommended.

Many initiators would want to use a different HI for different responders. The implementations should provide for a policy of initiator HIT to responder HIT. This policy should also include preferred transforms and local lifetimes.

Responders would need a similar policy, representing which hosts they accept HIP exchanges, and the preferred transforms and local lifetimes.

Moskowitz & Nikander Expires March 1, 2004 [Page 16]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## 9. Benefits of HIP

In the beginning, the network layer protocol (i.e. IP) had the following four "classic" invariants:

Non-mutable: The address sent is the address received.

Non-mobile: The address doesn't change during the course of an "association".

Reversible: A return header can always be formed by reversing the source and destination addresses.

Omniscient: Each host knows what address a partner host can use to send packets to it.

Actually, the fourth can be inferred from 1 and 3, but it is worth mentioning for reasons that will be obvious soon if not already.

In the current "post-classic" world, we are trying intentionally to get rid of the second invariant (both for mobility and for multi-homing), and we have been forced to give up the first and the fourth. Realm Specific IP [3] is an attempt to reinstate the fourth invariant without the first invariant. IPv6 is an attempt to reinstate the first invariant.

Few systems on the Internet have DNS names that are meaningful to them. That is, if they have a Fully Qualified Domain Name (FQDN), that typically belongs to a NAT device or a dial-up server, and does not really identify the system itself but its current connectivity. FQDN names (and their extensions as email names) are Application Layer names; more frequently naming processes than a particular system. This is why many systems on the internet are not registered in DNS; they do not have processes of interest to other Internet hosts.

DNS names are indirect references to IP addresses. This only demonstrates the interrelationship of the networking and application layers. DNS, as the Internet's only deployed, distributed, database is also the repository of other namespaces, due in part to DNSSEC and application specific key records. Although each namespace can be stretched (IP with v6, DNS with KEY records), neither can adequately provide for host authentication or act as a separation between internetworking and transport layers.

The Host Identity (HI) namespace fills an important gap between the IP and DNS namespaces. An interesting thing about the HI is that it actually allows one to give-up all but the 3rd Network Layer

Moskowitz & Nikander Expires March 1, 2004 [Page 17]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

invariant. That is to say, as long as the source and destination addresses in the network layer protocol are reversible, then things work ok because HIP takes care of host identification, and reversibility allows one to get a packet back to one's partner host. You don't care if the network layer address changes in transit (mutable) and you don't care what network layer address the partner is using (non-omniscient).

Since all systems can have a Host Identity, every system can have an entry in the DNS. The mobility features in HIP make it attractive to trusted 3rd parties to offer rendezvous servers.

## 9.1 HIP's Answers to NSRG questions

The IRTF Name Space Research Group has posed a number of evaluating questions in their report [6]. In this section, we provide answers to these questions.

### 1. How would a stack name improve the overall functionality of the Internet?

At the fundamental level, HI decouples the internetworking layer from the transport layer, allowing each to evolve separately. At the same time, the decoupling makes end-host mobility and multi-homing easier. It also allows mobility and multi-homing across the IPv4 and IPv6 networks. HIs make network renumbering easier. At the conceptual level, they also make process migration and clustered servers easier to implement. Furthermore, being cryptographic in nature, they provide the basis for solving the security problems related to end-host mobility and multi-homing.

### 2. What does a stack name look like?

A HI is a cryptographic public key. However, instead of using the keys directly, most protocols use a fixed size hash of the public key.

### 3. What is its lifetime?

HIP provides both stable and temporary Host Identifiers. Stable HIs are typically long lived, with a lifetime of years or more. The lifetime of temporary HIs depends on how long the upper layer connections and applications need them, and can range from a few seconds to years.

### 4. Where does it live in the stack?

Moskowitz & Nikander	Expires March 1, 2004	[Page 18]
□		
Internet-Draft	Host Identity Protocol Architecture	Sep 2003

The HIs live between the transport and internetworking layers.

### 5. How is it used on the end points

The Host Identifiers, in the form of HITs or LSIs, are used by legacy applications as if they were IP addresses. Additionally, the Host Identifiers, as public keys, are used in the built in key agreement protocol, called the HIP base exchange, to authenticate the hosts to each other.

### 6. What administrative infrastructure is needed to support it?

It is possible to use HIP opportunistically, without any infrastructure. However, to gain full benefit from HIP, the HIs must be stored in the DNS or a PKI, and a new infrastructure of rendezvous servers is needed.

7. If we add an additional layer would it make the address list in SCTP unnecessary?

Yes

8. What additional security benefits would a new naming scheme offer?

HIP reduces dependency on IP addresses, making the so called address ownership problems easier to solve. In practice, HIP provides security for end-host mobility and multi-homing. Furthermore, since HIP Host Identifiers are public keys, standard public key certificate infrastructures can be applied on the top of HIP.

9. What would the resolution mechanisms be, or what characteristics of a resolution mechanisms would be required?

For most purposes, an approach where DNS names are resolved simultaneously to HIs and IP addresses is sufficient. However, if it becomes necessary to resolve HIs into IP addresses or back to DNS names, a flat, hash based resolution infrastructure is needed. Such an infrastructure could be based on the ideas of Distributed Hash Tables, but would require significant new development and deployment.

Moskowitz & Nikander Expires March 1, 2004 [Page 19]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## 10. Security Considerations

HIP takes advantage of the new Host Identity paradigm to provide secure authentication of hosts and to provide a fast key exchange for IPsec. HIP also attempts to limit the exposure of the host to various denial-of-service (DoS) and man-in-the-middle (MitM) attacks. In so doing, HIP itself is subject to its own DoS and MitM attacks that potentially could be more damaging to a host's ability to conduct business as usual.

Resource exhausting Denial-of-service attacks take advantage of the cost of setting up a state for a protocol on the responder compared to the 'cheapness' on the initiator. HIP allows a responder to increase the cost of the start of state on the initiator and makes an effort to reduce the cost to the responder. This is done by having the responder start the authenticated Diffie-Hellman exchange instead of the initiator, making the HIP base exchange 4 packets long. There are more details on this process in the Host Identity Protocol specification [4].

HIP optionally supports opportunistic negotiation. That is, if a host receives a start of transport without a HIP negotiation, it can attempt to force a HIP exchange before accepting the connection.

This has the potential for DoS attacks against both hosts. If the method to force the start of HIP is expensive on either host, the attacker need only spoof a TCP SYN. This would put both systems into the expensive operations. HIP avoids this attack by having the responder send a simple HIP packet that it can pre-build. Since this packet is fixed and easily replayed, the initiator only reacts to it if it has just started a connection to the responder.

Man-in-the-middle attacks are difficult to defend against, without third-party authentication. A skillful MitM could easily handle all parts of the HIP base exchange, but HIP indirectly provides the following protection from a MitM attack. If the responder's HI is retrieved from a signed DNS zone or secured by some other means, the initiator can use this to authenticate the signed HIP packets. Likewise, if the initiator's HI is in a secure DNS zone, the responder can retrieve it and validate the signed HIP packets. However, since an initiator may choose to use an anonymous HI, it knowingly risks a MitM attack. The responder may choose not to accept a HIP exchange with an anonymous initiator.

In HIP, the Security Association for IPsec is indexed by the SPI; the source address is always ignored, and the destination address may be ignored as well. Therefore, HIP enabled IPsec Encapsulated Security Payload (ESP) is IP address independent. This might seem to make it easier for an attacker, but ESP with replay protection is already as

Moskowitz &amp; Nikander

Expires March 1, 2004

[Page 20]

Internet-Draft

Host Identity Protocol Architecture

Sep 2003

well protected as possible, and the removal of the IP address as a check should not increase the exposure of IPsec ESP to DoS attacks.

Since not all hosts will ever support HIP, ICMPv4 'Destination Unreachable, Protocol Unreachable' and ICMPv6 'Parameter Problem, Unrecognized Next Header' messages are to be expected and present a DoS attack. Against an initiator, the attack would look like the responder does not support HIP, but shortly after receiving the ICMP message, the initiator would receive a valid HIP packet. Thus, to protect against this attack, an initiator should not react to an ICMP message until a reasonable time has passed, allowing it to get the real responder's HIP packet. A similar attack against the responder is more involved.

Another MitM attack is simulating a responder's administrative rejection of a HIP initiation. This is a simple ICMP 'Destination Unreachable, Administratively Prohibited' message. A HIP packet is not used because it would either have to have unique content, and thus difficult to generate, resulting in yet another DoS attack, or just as spoofable as the ICMP message. Like in the previous case, the defense against this attack is for the initiator to wait a reasonable time period to get a valid HIP packet. If one does not come, then the initiator has to assume that the ICMP message is valid. Since this is the only point in the HIP base exchange where this ICMP message is appropriate, it can be ignored at any other point in the exchange.

#### 10.1 HITs used in ACLs

It is expected that HITs will be used in ACLs. Future firewalls can use HITs to control egress and ingress to networks, with an assurance level difficult to achieve today. As discussed above in Section 6, once a HIP session has been established, the SPI value in an IPsec packet may be used as an index, indicating the HITs. In practise, the firewalls can inspect the HIP packets to learn of the bindings

between HITs, SPI values, and IP addresses. They can even explicitly control IPsec usage, dynamically opening IPsec ESP only for specific SPI values and IP addresses. The signatures in the HIP packets allow a capable firewall to make sure that the HIP exchange is indeed happening between two known hosts. This may increase firewall security.

There has been considerable bad experience with distributed ACLs that contain public key related material, for example, with SSH. If the owner of the key needs to revoke it for any reason, the task of finding all locations where the key is held in an ACL may be impossible. If the reason for the revocation is due to private key theft, this could be a serious issue.

Moskowitz & Nikander Expires March 1, 2004 [Page 21]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

A host can keep track of all of its partners that might use its HIT in an ACL by logging all remote HITs. It should only be necessary to log responder hosts. With this information, the host can notify the various hosts about the change to the HIT. There has been no attempt to develop a secure method (like in CMP and CMC) to issue the HIT revocation notice.

NATs, however, are transparent to the HIP aware systems by design. Thus, the host may find it difficult to notify any NAT that is using a HIT in an ACL. Since most systems will know of the NATs for their network, there should be a process by which they can notify these NATs of the change of the HIT. This is mandatory for systems that function as responders behind a NAT. In a similar vein, if a host is notified of a change in a HIT of an initiator, it should notify its NAT of the change. In this manner, NATs will get updated with the HIT change.

## 10.2 Non-security Considerations

The definition of the Host Identifier states that the HI need not be a public key. It implies that the HI could be any value; for example an FQDN. This document does not describe how to support such a non-cryptographic HI. A non-cryptographic HI would still offer the services of the HIT or LSI for NAT traversal. It would be possible carry the HITs in HIP packets that had neither privacy nor authentication. Since such a mode would offer so little additional functionality for so much addition to the IP kernel, it has not been defined. Given how little public key cryptography HIP requires, HIP should only be implemented using public key Host Identities.

If it is desirable to use HIP in a low security situation where public key computations are considered expensive, HIP can be used with very short Diffie-Hellman and Host Identity keys. Such use makes the participating hosts vulnerable to MitM and connection hijacking attacks. However, it does not cause flooding dangers, since the address check mechanism relies on the routing system and not on cryptographic strength.

Moskowitz & Nikander Expires March 1, 2004 [Page 22]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## 11. Acknowledgments

For the people historically involved in the early stages of HIP, see the Acknowledgements section in the Host Identity Protocol specification [4].

During the later stages of this document, when the editing baton was transferred to Pekka Nikander, the comments from the early implementors and others, including Jari Arkko, Tom Henderson, Petri Jokela, Miika Komu, Mika Kousa, Andrew McGregor, Jan Melen, Tim Shepard, Jukka Ylitalo, and Jorma Wall, were invaluable.

Moskowitz & Nikander Expires March 1, 2004 [Page 23]  
 □  
 Internet-Draft Host Identity Protocol Architecture Sep 2003

## References (informative)

[1] Tsirtsis, G. and P. Srisuresh, "Network Address Translation -

... /web/20031212192243/http://www.ietf.org/internet-drafts/draft-moskowi... 18-10-2004  
 BNSDOCID: <XP\_\_\_\_\_2300908A\_1\_>

Protocol Translation (NAT-PT)", RFC 2766, February 2000.

- [2] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [3] Borella, M., Lo, J., Grabelsky, D. and G. Montenegro, "Realm Specific IP: Framework", RFC 3102, October 2001.
- [4] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", draft-moskowitz-hip-07 (work in progress), June 2003.
- [5] Richardson, M., "A method for storing IPsec keying material in DNS", draft-ietf-ipseckey-rr-07 (work in progress), September 2003.
- [6] Lear, E. and R. Droms, "What's In A Name: Thoughts from the NSRG", draft-irtf-nsrg-report-10 (work in progress), September 2003.
- [7] Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-nikander-hip-mm-00 (work in progress), June 2003.
- [8] Nikander, P., "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-01 (work in progress), July 2003.
- [9] Chiappa, J., "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture", URL <http://users.exis.net/~jnc/tech/endpoints.txt>, 1999.

#### Authors' Addresses

Robert Moskowitz  
 ICSAlabs, a Division of TruSecure Corporation  
 1000 Bent Creek Blvd, Suite 200  
 Mechanicsburg, PA  
 USA

EMail: [rgm@icsalabs.com](mailto:rgm@icsalabs.com)

Moskowitz & Nikander	Expires March 1, 2004	[Page 24]
Internet-Draft	Host Identity Protocol Architecture	Sep 2003

Pekka Nikander  
 Ericsson Research Nomadic Lab

JORVAS FIN-02420  
 FINLAND

Phone: +358 9 299 1  
 EMail: [pekka.nikander@nomadiclab.com](mailto:pekka.nikander@nomadiclab.com)



Moskowitz & Nikander Expires March 1, 2004 [Page 25]  
□  
Internet-Draft Host Identity Protocol Architecture Sep 2003

#### Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

#### Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION

Moskowitz & Nikander	Expires March 1, 2004	(Page 26)
Internet Draft	Host Identity Protocol Architecture	Sep 2003

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Moskowitz & Nikander

Expires March 1, 2004

[Page 27]

□